



Synthèse et validation de lois de commande de vol robustes en présence d'incertitudes paramétriques et de non-linéarités

J. Lesprier

► To cite this version:

J. Lesprier. Synthèse et validation de lois de commande de vol robustes en présence d'incertitudes paramétriques et de non-linéarités. Physique de l'espace [physics.space-ph]. UNIVERSITE DE TOULOUSE, 2015. Français. NNT: . tel-01247970

HAL Id: tel-01247970

<https://hal.science/tel-01247970>

Submitted on 23 Dec 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Institut Supérieur de l'Aéronautique et de l'Espace (ISAE)*

Présentée et soutenue le 3 décembre 2015 par :

JÉRÉMY LESPRIER

**Synthèse et validation de lois de commande de vol robustes en présence
d'incertitudes paramétriques et de non-linéarités.**

JURY

M. JAMAL DAAFOUZ
M. GÉRARD SCORLETTI
M. STÉPHANE DELANNOY
M. DIMITRI PEAUCELLE
M. JEAN-MARC BIANNIC
M. CLÉMENT ROOS

Professeur des Universités
Professeur des Universités
Ingénieur
Directeur de Recherche
Ingénieur de Recherche
Ingénieur de Recherche

Rapporteur
Rapporteur
Examineur
Président du jury
Directeur de thèse
Co-Directeur de thèse

École doctorale et spécialité :

EDSYS : Automatique 4200046

Unité de Recherche :

ONERA - The French Aerospace Lab - Commande des Systèmes et Dynamique du Vol

Directeur(s) de Thèse :

M. JEAN-MARC BIANNIC et M. CLÉMENT ROOS

Rapporteurs :

M. JAMAL DAAFOUZ et M. GÉRARD SCORLETTI

REMERCIEMENTS

Lorsque vient le moment de rédiger la partie dédiée aux remerciements, c'est qu'un long chemin vient d'être parcouru. Ce moment signe d'une part la fin de trois années passionnantes et enrichissantes qui se sont déroulées à l'ONERA de Toulouse, au département "Dynamique des systèmes et commande de vol". Et d'autre part, il annonce également l'approche de nouveaux horizons, et le commencement d'une nouvelle vie professionnelle, amenant d'autres contraintes et d'autres méthodes de fonctionnement.

Mais pour en revenir sur ces trois années, l'idée au départ de mêler des méthodes d'automatique, l'analyse de robustesse, et des applications aéronautiques ne pouvait que me séduire techniquement. Il faut cependant souligner que le bon déroulement de cette thèse – comprenant entre autres l'acquisition des connaissances, l'élaboration de nouveaux résultats, la rédaction de papiers et du mémoire, jusqu'à l'obtention du diplôme – ne peut être décorrélé des personnes qui m'ont entouré durant tout ce temps. C'est pourquoi cette section leur est particulièrement adressée, et je vais veiller à n'oublier personne, même si ce n'est pas une mince affaire !

Tout d'abord, je tiens à remercier les directeurs successifs du département DCSD, Patrick Fabiani au départ, puis Bruno Lamiscarre, pour avoir permis à cette thèse de se dérouler au sein de l'ONERA. J'adresse mes remerciements à Christelle Cumer, chef de l'unité CDIN (Commande et Intégration), pour l'accueil et la gestion de l'unité. Pour le côté logistique, permettant notamment d'assurer les démarches administratives, je remercie également grandement la secrétaire du département, Valérie Cassagnol, qui m'a été d'une grande aide durant ces trois ans.

Puis, je souhaite exprimer toute ma gratitude à mon directeur et à mon co-directeur de thèse, respectivement Jean-Marc Biannic et Clément Roos. Vos compétences techniques complémentaires m'ont permis d'acquérir les connaissances nécessaires pour la réussite de ces travaux de thèse. Vous avez toujours été d'un immense soutien, tant humain que scientifique, et m'avez accordé une grande confiance, me permettant ainsi d'avancer et d'arriver à atteindre une partie des objectifs fixés (on peut toujours faire plus !). Le bon déroulement de cette thèse vous est donc largement dû... Encore merci pour tout.

Je tiens à remercier Jamal Daafouz (Professeur au CRAN (Nancy)) et Gérard Scorletti (Professeur à l'Ecole Centrale de Lyon), pour avoir accepté d'être les rapporteurs de ma thèse et pour leurs retours pertinents sur le manuscrit, Dimitri Peaucelle (Directeur de Recherche au LAAS-CNRS (Toulouse)) pour avoir présidé le jury, et Stéphane Delannoy (Ingénieur chez AIRBUS (Toulouse)) pour avoir participé au jury avec un point de vue industriel fort pertinent sur ces travaux. J'ai été honoré de présenter mon travail devant ces membres de jury, qui ont su exprimer leur intérêt, tout en introduisant des remarques et conseils avisés.

Même si malheureusement le projet n'a pas (encore) totalement abouti sur le plan implémentation et essais, je remercie également toute l'équipe du laboratoire drones, dont Pierre Escalas, Mathieu Rognant, et Alexandre Amiez. Mes pensées vont de plus envers les proches de Paul Chavent, dont la disparition m'a beaucoup affecté, et qui s'est avéré être d'une très grande aide pour la modélisation numérique d'un drone à voilure fixe.

Ensuite, vient la longue liste des collègues rencontrés à l'ONERA, qui ont de près ou de loin contribué à la bonne ambiance qui régnait au sein du département, et avec qui j'ai pu partager des discussions scientifiques et techniques ainsi que d'autres très bons moments. Je commence tout d'abord par certains permanents, dont Charles P-V., Cédric S., et Henry D.P. Vient ensuite la liste des stagiaires et doctorants, avec qui j'ai pu partager mon bureau : Jérémy V., Aurélie T., Gabiche, Moumoune, et particulièrement Jorrit T. et Marine H. qui ont su me supporter durant la rédaction du mémoire, période pas toujours la plus évidente ! Et tous les autres : Manu, Adrien M., Mathieu B., Patrick B., Jââcques, Nico D., Francis, Hélène E., Alvaro P., Igor P.D.P., Martin S., Guillaume A., Victor G., Guillaume C., Adèle B., Lylia S., Thomas S., Marie D., Pierre V., Henri S., Razvan L., Elodie D., Simon O., Nicolas G., Yann A., Gaëtan S, et Jean-Philippe C.

Enfin et plus personnellement, je tiens à remercier mes amis (notamment Mathieu C., David H., Julien B., et Clémence B.) pour leur soutien moral et leur intérêt. Et je compte adresser un remerciement tout particulier pour mes parents, qui ont toujours été là pour me soutenir, et à qui je dois une grande partie de ma réussite tout au long de mes études, ainsi qu'à mes frères.

Et pour clore cette section, un immense merci à un ma bien-aimée Tiphaine qui m'a soutenu tout le long, malgré la distance parfois, et qui m'a apporté motivation et m'a permis de me ressourcer dans les moments où j'en avais le plus besoin.

Un *grand* merci à tous !

TABLE DES MATIÈRES

TABLE DES MATIÈRES	v
LISTE DES FIGURES	ix
NOTATIONS	xiii
INTRODUCTION GÉNÉRALE	1
I Analyse de robustesse des systèmes LTI	9
1 NOTIONS THÉORIQUES SUR L'ANALYSE DE ROBUSTESSE DES SYSTÈMES LTI	11
1.1 MODÉLISATION LFT ET INCERTITUDES STRUCTURÉES	13
1.1.1 Quelques rappels sur la transformation linéaire fractionnaire . . .	13
1.1.2 Théorème du faible gain et structure des incertitudes	17
1.2 VALEUR SINGULIÈRE STRUCTURÉE ET μ -ANALYSE	18
1.2.1 Introduction à la valeur singulière structurée	18
1.2.2 Considérations sur la borne inférieure de μ	19
1.2.3 Borne supérieure de μ : approche par matrices (D,G)	25
1.2.4 Borne supérieure de μ : autres approches	29
1.3 DU CONSERVATISME DE LA BORNE SUPÉRIEURE DE μ	35
CONCLUSION	38
2 TECHNIQUES DE RÉDUCTION DU CONSERVATISME DE LA BORNE SUPÉRIEURE DE μ	39
2.1 INTRODUCTION AUX μ -SENSIBILITÉS	42
2.1.1 Définition des μ -sensibilités	42
2.1.2 Considérations numériques	43
2.2 AMÉLIORATION DES APPROCHES BASÉES SUR LA FORMULATION PAR MATRICES (D,G)	48
2.2.1 Résolution partielle des LMIs de la caractérisation par matrices (D,G)	48
2.2.2 Nouvel algorithme de branch-&-bound	51

2.3	UNE BORNE SUPÉRIEURE DE μ MOINS CONSERVATIVE PAR L'APPROCHE PAR MULTIPLIEURS	53
2.3.1	Rappels de l'approche	54
2.3.2	μ -sensibilités et réduction du temps de calcul	57
2.3.3	Validation des multiplieurs sur de larges intervalles fréquentiels	59
2.3.4	Résultats	63
	CONCLUSION	66
II	Synthèse de lois de commande non-linéaires robustes	67
3	INTRODUCTION À L'INVERSION DYNAMIQUE NON-LINÉAIRE	69
3.1	CAS DES SYSTÈMES CARRÉS AFFINES EN LA COMMANDE	71
3.1.1	Principe général	72
3.1.2	Limitations	73
3.2	EXTENSION : TECHNIQUE DE LINÉARISATION ENTRÉE-SORTIE	74
3.2.1	Cas des systèmes non-linéaires SISO	74
3.2.2	Systèmes avec échelles de temps multiples	78
3.2.3	Systèmes non-linéaires MIMO	80
3.3	TECHNIQUE DE ROBUSTIFICATION : SYNTHÈSE \mathcal{H}_∞	83
3.3.1	La problème \mathcal{H}_∞ standard	83
3.3.2	Quelques notes sur les fonctions de pondération	85
3.3.3	La synthèse \mathcal{H}_∞ structurée	87
	CONCLUSION	89
4	TECHNIQUE DE COMPENSATION NON-LINÉAIRE ROBUSTE	91
4.1	DES SYSTÈMES AFFINES EN LA COMMANDE ET DÉPENDANT DE PARAMÈTRES	94
4.1.1	Description du système	94
4.1.2	Notations et hypothèses	95
4.2	UNE NOUVELLE APPROCHE BASÉE SUR L'INVERSION DYNAMIQUE ROBUSTE	96
4.2.1	Boucle interne de linéarisation	96
4.2.2	Boucle externe de robustification	97
4.3	MODÉLISATION LFT ET ANALYSE DE ROBUSTESSE	100
4.3.1	Incertitudes de modélisation	101
4.3.2	Incertitudes au niveau des actionneurs	101
4.3.3	Incertitudes sur les perturbations mesurées	102
4.3.4	Forme LFT standard du système bouclé	102
4.4	SYNTHÈSE MULTI-MODÈLES	103
4.5	VERS UNE ANALYSE GLOBALE	106
	CONCLUSION	108

III Applications aéronautiques	111
5 ÉQUATIONS GÉNÉRALES DE LA MÉCANIQUE DU VOL	113
5.1 NOTATIONS PHYSIQUES ET DÉFINITION DES REPÈRES	115
5.1.1 Notations physiques	115
5.1.2 Définition des repères	116
5.2 ÉQUATIONS GÉNÉRALES	119
5.2.1 Équations de la dynamique	119
5.2.2 Équations de la cinématique	120
5.3 FORCES ET MOMENTS	121
5.3.1 Expression des forces	121
5.3.2 Expression des moments	122
5.4 MODÈLE ATMOSPHÉRIQUE ET TURBULENCES	124
5.5 COEFFICIENTS AÉRODYNAMIQUES	126
CONCLUSION	127
6 APPLICATION À L'ATTERRISSAGE AUTOMATIQUE D'UN AVION DE LIGNE	129
6.1 SYNTHÈSE DU CORRECTEUR LONGITUDINAL	131
6.1.1 Modèle dynamique et objectifs	131
6.1.2 Application de la méthode de compensation non-linéaire	133
6.1.3 Analyse de robustesse et synthèse multi-modèles	136
6.2 SYNTHÈSE DU CORRECTEUR LATÉRAL	142
6.2.1 Modèle dynamique et objectifs	142
6.2.2 Application de la méthode complète et simulations	144
6.3 GUIDAGE POUR L'ATTERRISSAGE AUTOMATIQUE PAR SUIVI D'ILS	146
6.3.1 ILS et guidage	147
6.3.2 Phases d'arrondi et de décroche	149
6.3.3 Simulations temporelles	150
CONCLUSION	155
7 MODÉLISATION ET PILOTAGE ROBUSTE D'UN DRONE À VOILURE FIXE	157
7.1 DESCRIPTION DU DRONE "AVION JAUNE"	159
7.2 MODÉLISATION NUMÉRIQUE	160
7.2.1 Détermination des paramètres structuraux	160
7.2.2 Détermination des coefficients aérodynamiques	161
7.2.3 Modélisation des élévons	163
7.2.4 Modélisation du bloc de propulsion	164
7.2.5 Simulations en boucle ouverte et comparaison avec des données de vol	167
7.3 SYNTHÈSE DE LOIS DE PILOTAGE AUTOMATIQUE	173

7.3.1	Synthèse modale	173
7.3.2	Technique de compensation non-linéaire robuste	174
7.3.3	Simulations temporelles	176
7.4	DE L'IMPLANTATION RÉELLE DES LOIS	177
	CONCLUSION	178
	CONCLUSION GÉNÉRALE	181
	PUBLICATIONS	187
	BIBLIOGRAPHIE	189

LISTE DES FIGURES

1.1	LFTs inférieure (gauche) et supérieure (droite).	14
1.2	Interconnexion standard pour l'analyse de stabilité robuste.	17
1.3	Schéma d'interconnexion.	32
3.1	Problème \mathcal{H}_∞ standard	83
3.2	Schéma de synthèse \mathcal{H}_∞	86
3.3	Synthèse \mathcal{H}_∞ multi-modèles.	88
4.1	Schéma de synthèse \mathcal{H}_∞ multi-objectifs.	99
4.2	Schéma d'analyse de stabilité robuste de la boucle fermée.	102
4.3	Forme LFT du système bouclé pour l'analyse de robustesse.	103
5.1	Repère avion et description des diverses notations angulaires.	117
5.2	Modèle Simulink générique d'un avion civil pour la simulation et la commande, présent dans la boîte à outils Matlab SMAC [SMAC 2012-2015].	128
6.1	Simulations sur le modèle non-linéaire complet, avec un échelon de 3 degrés sur l'angle de vol γ , pour différentes configurations paramétriques du domaine admissible Θ , et aucune incertitude de modélisation, avec le correcteur nominal $\hat{K}_0(s)$	136
6.2	Simulations sur le modèle non-linéaire complet, avec un échelon de 3 m/s sur la vitesse air V_a , pour différentes configurations paramétriques du domaine admissible Θ , et aucune incertitude de modélisation, avec le correcteur nominal $\hat{K}_0(s)$	137
6.3	Simulations sur le modèle non-linéaire complet, avec un échelon de 3 degrés sur l'angle de vol γ , pour différentes configurations paramétriques du domaine admissible Θ , et aucune incertitude de modélisation, avec le correcteur multi-modèles $\hat{K}(s)$	139
6.4	Simulations sur le modèle non-linéaire complet, avec un échelon de 3 m/s sur la vitesse air V_a , pour différentes configurations paramétriques du domaine admissible Θ , et aucune incertitude de modélisation, avec le correcteur multi-modèles $\hat{K}(s)$	140

6.5	Comparaison entre le correcteur nominal $\hat{K}_0(s)$ (a) et le correcteur multi-modèles $\hat{K}(s)$ (b), pour toutes les configurations du domaine admissible Θ avec $\pm 25\%$ d'incertitudes de modélisation sur les coefficients aérodynamiques.	141
6.6	Simulations sur le modèle non-linéaire complet, avec un échelon de 10 degrés sur l'angle de roulis ϕ , pour différentes configurations paramétriques du domaine admissible Θ , et $\pm 30\%$ d'incertitudes sur les coefficients aérodynamiques latéraux, avec le correcteur multi-modèles $\hat{K}(s)$	145
6.7	Simulations sur le modèle non-linéaire complet, avec un échelon de 10 degrés sur l'angle de dérapage β , pour différentes configurations paramétriques du domaine admissible Θ , et $\pm 30\%$ d'incertitudes sur les coefficients aérodynamiques latéraux, avec le correcteur multi-modèles $\hat{K}(s)$	146
6.8	Signal "localizer" en vue de la correction de la déviation latérale de l'avion par rapport à l'axe de la piste d'atterrissage.	147
6.9	Signal "glide" en vue de la correction verticale de l'avion par rapport à la pente optimale de descente pour l'atterrissage.	148
6.10	Schéma global de commande et de guidage pour l'atterrissage automatique.	149
6.11	Simulations d'atterrissage complet en présence de vent latéral, pour différentes configurations de l'avion.	152
6.12	Interpolations gaussiennes et probabilités qu'une valeur atteigne le seuil critique après 10000 simulations de Monte Carlo : exigences longitudinales.	153
6.13	Interpolations gaussiennes et probabilités qu'une valeur atteigne le seuil critique après 10000 simulations de Monte Carlo : exigences latérales.	154
7.1	Drone aérien à voilure fixe, l'Avion Jaune.	160
7.2	Modélisation de l'Avion Jaune sur CATIA, en vue de l'obtention numérique de la matrice d'inertie (les valeurs étant cachées par souci de confidentialité).	161
7.3	Représentation 3D réalisée sous XFLR5 du drone à voilure fixe. . . .	162
7.4	Comparaison entre les données expérimentales et le modèle des actionneurs des élévons suite à une sollicitation en rampe.	164
7.5	Comparaison entre les données expérimentales (bleu) et le modèle (rouge) pour la vitesse moteur R_m	165

7.6	Évolution de la vitesse de rotation du moteur (rpm) en fonction de la vitesse air V_a et de la commande de poussée δ_{xf}	165
7.7	Comparaison de la poussée statique ($V_a = 0$) expérimentale (bleu) et modélisée (rouge).	166
7.8	Données expérimentales permettant de valider partiellement le modèle de poussée en vol. Les points rouges correspondent à des phases de vol descendantes, les points verts à des phases ascendantes, dont les premiers instants sont mis en évidence par des cercles noirs, et le trait noir continu à la zone de poussée nulle.	167
7.9	Évolution de la poussée moteur en fonction de la vitesse de rotation et de la vitesse air.	168
7.10	Comparaison sur 30 secondes de données simulées et de vol : modèle longitudinal.	169
7.11	Comparaison sur 30 secondes de données simulées et de vol : modèle latéral.	170
7.12	Comparaison de données simulées et de vol lors d'un braquage de la profondeur : modèle longitudinal.	171
7.13	Comparaison de données simulées et de vol lors d'un braquage de la profondeur : modèle latéral.	171
7.14	Comparaison de données simulées et de vol lors d'un braquage des ailerons : modèle longitudinal.	172
7.15	Comparaison de données simulées et de vol lors d'un braquage des ailerons : modèle latéral.	173
7.16	Simulations sur le modèle non-linéaire complet pour une consigne échelon de 5 degrés sur γ , avec le correcteur modal (ligne en tirets bleus) et la loi non-linéaire robuste multi-modèles (trait plein rouge), pour la configuration nominale sans incertitudes	176
7.17	Simulations sur le modèle non-linéaire complet pour consigne échelon de 5 degrés sur γ , avec le correcteur modal (ligne en tirets bleus) et la loi non-linéaire robuste multi-modèles (trait plein rouge), pour la configuration pire-cas avec incertitudes	177

NOTATIONS

BMI	Inégalités Matricielles Bilinéaires
QIC	Contraintes Quadratiques Intégrales
LFR	Représentation Linéaire Fractionnaire
LFT	Transformation Linéaire Fractionnaire
LMI	Inégalité Matricielle Linéaire
LPV	Linéaire à Paramètres Variants
LTi	Linéaire à Temps Invariant
LTV	Linéaire à Temps Variant
NDI	Inversion dynamique non-linéaire
SISO	Mono-entrée, mono-sortie
MIMO	Multi-entrées, multi-sorties
\mathbb{R}	Ensemble des nombres réels
\mathbb{C}	Ensemble des nombres complexes
\mathbb{N}	Ensemble des entiers naturels
$\mathcal{F}_l(\cdot)$	Transformation Linéaire Fractionnaire inférieure
$\mathcal{F}_u(\cdot)$	Transformation Linéaire Fractionnaire supérieure
I_n	Matrice identité de taille n
x^T	Transposée du vecteur x
M^T	Transposée de la matrice M
M^*	Transconjugée de la matrice M
$\bar{\sigma}(M)$	Plus grande valeur singulière de la matrice M
$\bar{\lambda}(M)$	Plus grande valeur propre de la matrice M
$\mathcal{RL}_\infty^{n \times n}$	Espace constitué des fonctions de transfert propres, à coefficients réels, de taille $n \times n$ et sans pôle sur l'axe imaginaire
$\mathcal{RH}_\infty^{n \times n}$	Sous-espace de $\mathcal{RL}_\infty^{n \times n}$ constitué des fonctions sans pôle dans le demi-plan droit
$\ M(s)\ _\infty$	Norme \mathcal{H}_∞ de la fonction de transfert $M(s) \in \mathcal{RH}_\infty^{n \times n}$
$\text{Re}(\cdot)$	Partie réelle
$\text{Im}(\cdot)$	Partie imaginaire

INTRODUCTION GÉNÉRALE

CONTEXTE

DEPUIS l'apparition des systèmes de commande de vol électrique à bord des avions – que l'on peut situer au milieu des années 1980 – le développement de méthodes de synthèse de lois de pilotage automatique s'est avéré essentiel. Il a été motivé par une volonté – toujours actuelle – d'améliorer les performances en vol de l'avion, tout en assurant une sécurité accrue pour ses occupants et en permettant aux pilotes de se focaliser principalement sur la navigation. Le concepteur de ces lois doit accorder autant d'importance à la phase de modélisation, qu'à la méthode utilisée pour les élaborer, ainsi qu'à leur validation. Il se confronte alors à un certain nombre de problématiques rendant leur synthèse délicate. Tout d'abord, un aéronef évolue généralement dans un grand domaine de fonctionnement qui impose une adaptation des lois au point de vol considéré (dépendant par exemple de l'altitude ou du nombre de Mach). De plus, la modélisation contient de nombreuses sources d'incertitudes qui, si elles ne sont pas ou peu prises en considération, peuvent impacter de façon négative le comportement général de l'avion en vol. Il en résulte un nombre d'itérations conséquent afin de recalibrer les modèles et ajuster les lois de pilotage, en fonction des résultats d'essais en vol. Ce processus peut s'avérer alors coûteux en temps et en ressources.

Du pilotage robuste sur un large domaine d'évolution

Pour pallier le large domaine d'évolution et permettre un pilotage robuste de l'appareil, la technique standard de séquençage de gains (voir par exemple [Shahruz et Behtash 1990; Shamma et Athans 1990]) permet de dimensionner plusieurs lois de pilotage à différents points de vol. Ceci fournit un ensemble de lois de pilotage linéaires qui, une fois interpolées, couvrent tout le domaine de vol. Même si cette technique est encore de nos jours très utilisée dans l'industrie aéronautique, elle demande d'une part un temps de développement très long du fait des nombreux modèles linéarisés à traiter, et d'autre part elle pose un souci de vérification de stabilité entre les points de fonctionnement. Seules l'expérience et des vérifications par simulations de Monte-Carlo, ainsi que de nombreux essais en

vol, permettent de garantir – de façon empirique donc – la validité de ces lois. Il est d’ailleurs parfois extrêmement délicat d’obtenir une bonne interpolation des correcteurs synthétisés, notamment quand ces derniers sont d’ordre élevé.

Il existe de nombreuses alternatives aux techniques de séquençement de gains que nous venons d’évoquer. Parmi ces alternatives, on peut citer les techniques de synthèse LPV (Linéaire à Paramètres Variants) [Becker et al. 1993; Packard 1994; Apkarian et Gahinet 1995; Biannic et Apkarian 1995], dont la plupart reposent sur une modélisation générique sous forme LFT (Transformation Fractionnaire Linéaire) parfois conservative et difficile à obtenir, ou bien sur des modèles polytopiques [Daafouz 2008].

Dans le domaine de la commande non-linéaire, la technique de linéarisation entrée-sortie robuste (voir par exemple [Isidori 1985; Slotine et Li 1991; Vidyasagar 1993]), aussi appelée communément inversion dynamique, semblent être une alternative intéressante aux techniques classiques de commande LPV. Elle consiste à obtenir un modèle linéaire générique en utilisant une loi de commande qui va annuler partiellement les non-linéarités du système. Ceci permet non seulement de découpler une grande classe de modèles non-linéaires, mais aussi d’adapter la loi automatiquement au point de fonctionnement. En la couplant à une méthode de synthèse linéaire robuste, comme la synthèse \mathcal{H}_∞ , elle permet d’apporter une solution alternative très intéressante aux techniques de séquençement de gains, et a montré ses bénéfices dans de nombreuses applications aéronautiques (voir par exemple [Snell et al. 1992; Adams et Banda 1993; Reiner et al. 1995; 1996; Kara Mohamed et Lanzon 2012; Biannic et al. 2014]). Cependant, elle manque encore de robustesse vis à vis des dynamiques négligées ou des incertitudes de modélisation, qui doivent d’une façon ou d’une autre être prises en compte lors de la synthèse. Des solutions sont proposées dans ces travaux de thèse, notamment par l’évaluation de leur impact et par l’analyse de robustesse qui en découle.

De la prise en compte des incertitudes de modélisation et des dynamiques négligées

L’évaluation de l’impact des dynamiques négligées et des incertitudes de modélisation sur le comportement général de l’avion est ainsi essentielle à l’élaboration et à la validation des lois de pilotage. L’analyse de stabilité robuste des systèmes linéaires et non-linéaires repose sur de nombreuses théories qui ne seront pas toutes exposées ici. On s’intéressera particulièrement aux cas d’incertitudes linéaires à temps invariant (LTI) qui suffisent à représenter des incertitudes de modélisation ou des dynamiques négligées. La μ -analyse, introduite par [Doyle 1982], permet d’évaluer la marge de robustesse – par calcul de la valeur singulière structurée

μ – d'un système vis-à-vis de telles incertitudes. Évitant ainsi des simulations de Monte Carlo longues et coûteuses, elle constitue une technique intéressante pour analyser la stabilité robuste d'un système commandé. Le calcul exact de la valeur singulière structurée μ est cependant NP-difficile (voir [Braatz et al. 1994]). Seules des bornes inférieures et supérieures sont calculées. Les techniques de calcul de bornes inférieures de μ sont aujourd'hui très matures et ont été éprouvées sur des applications très complexes [Ferrerres et Biannic 2001; Roos et Biannic 2015]. Dans nos travaux, nous allons donc principalement nous focaliser sur les techniques de calcul de bornes supérieures.

Dans [Fan et al. 1991; Young et al. 1992; 1995] est développée la formulation dite des matrices (D,G). Elle permet de calculer une borne supérieure de μ avec un temps de calcul très raisonnable notamment par l'utilisation d'une technique de descente de gradient, ce qui la rend très intéressante pour des applications réalistes. Cependant, il arrive dans certains cas que la borne obtenue soit très conservative, i.e. très éloignée de la valeur exacte de μ , rendant l'analyse de robustesse imprécise. L'utilisation d'outils numériques de résolution de LMIs peut parfois pallier ce problème (voir [Beck et Doyle 1992]). Cependant, elle devient coûteuse en temps de calcul si le nombre d'incertitudes de modélisation augmente. De plus la formulation par matrices (D,G) peut tout simplement être la cause de l'excès de conservatisme de la borne supérieure de μ , montrant alors ses limites.

Dans [Newlin et Young 1997] est exposée une technique couplant la formulation par matrices (D,G) avec un algorithme de branch-&-bound qui permet théoriquement de réduire l'écart entre les bornes inférieure et supérieure de μ à une tolérance fixée par l'utilisateur. Là encore, bien que la technique soit théoriquement attractive, le temps de calcul augmente de façon exponentielle avec le nombre d'incertitudes présentes dans le système à analyser, d'autant plus si l'écart souhaité entre les bornes est faible.

L'approche par multiplieurs, développée dans [Fu et Barabanov 1997], généralise quant à elle l'approche par matrices (D,G). Elle est moins conservative, mais nécessite la résolution d'un système de LMIs pour calculer une borne supérieure, ce qui est problématique encore une fois pour le temps de calcul. En effet, le nombre de LMIs à résoudre croît exponentiellement avec le nombre d'incertitudes dans le système, ce qui rend cette méthode peu attractive pour des problèmes réalistes.

Une autre alternative, proposée dans [Scorletti et al. 2007], consiste à généraliser la caractérisation par matrices D et G par l'ajout d'une troisième matrice L, tout en conservant une complexité polynomiale de résolution.

Enfin, d'autres alternatives existent et seront évoquées dans cette thèse.

Résumé du contexte de la thèse

En résumé, le pilotage automatique robuste d'un avion sur un large domaine de vol est un sujet délicat. Les techniques actuelles nécessitent généralement un temps d'élaboration très long, et peuvent s'avérer peu robustes aux incertitudes de modélisation et aux dynamiques négligées, ces dernières n'étant pas ou peu prises en compte lors de la synthèse. L'évaluation de la robustesse vis-à-vis de ces incertitudes est essentielle pour pallier les défauts précédents. Celle-ci peut être réalisée par des approches déterministes, mais l'excès de conservatisme ou bien le temps de calcul nécessaire freinent son intégration dans un processus global de synthèse de lois.

Ainsi, les deux problématiques de synthèse de lois de pilotage et d'analyse de robustesse font partie intégrante des objectifs de cette thèse.

OBJECTIFS ET DÉMARCHE

Étant données les deux problématiques principales introduites précédemment, les objectifs de cette thèse sont les suivants :

- Développer une technique de synthèse robuste, suffisamment générique, pour le pilotage d'un aéronef sur l'ensemble de son domaine de vol, en présence d'incertitudes de modélisation et de variations paramétriques,
- Améliorer les techniques existantes d'analyse de robustesse par μ -analyse, afin d'évaluer une borne supérieure de μ moins conservative – et donc une marge de robustesse plus précise – tout en assurant un temps de calcul raisonnable,
- Appliquer les techniques développées à l'atterrissage automatique d'un avion civil, ainsi qu'au pilotage automatique d'un drone à voilure fixe.

Afin de répondre à ces objectifs, la thèse s'est articulée autour de deux grands axes, qui sont présentés dans ce mémoire selon un ordre jugé le plus adapté pour la compréhension du lecteur.

Dans un premier temps les travaux ont consisté à développer et à valider sur une liste de benchmarks réalistes des techniques de réduction du conservatisme et du temps de calcul d'une borne supérieure de μ . Pour ce faire, trois méthodes ont été développées :

- résolution partielle de LMIs dans le cadre de la caractérisation par matrices (D,G), afin de réduire le temps de calcul en présence d'un grand nombre d'incertitudes,

- utilisation de l’approche par multiplieurs, introduite par [Fu et Barabanov 1997], sur un nombre limité d’incertitudes, avec une méthode de validation de la borne et du multiplieur obtenus sur de larges intervalles fréquentielles,
- nouvel algorithme de branch-&-bound, où le domaine d’incertitudes est subdivisé non plus selon sa plus grande dimension mais selon la dimension correspondant à l’incertitude la plus influente sur la valeur de la borne supérieure de μ .

Toutes ces méthodes utilisent les μ -sensibilités pour sélectionner les incertitudes qui ont le plus d’impact sur la valeur de la borne supérieure de μ , réduisant ainsi la taille des LMIs à résoudre, ou permettant de subdiviser de manière pertinente le domaine d’incertitudes. Ceci permet d’améliorer sensiblement le temps de calcul. Ces développements ont donné lieu à deux publications [Lesprier et al. 2015c] [Lesprier et al. 2015b].

Les résultats obtenus permettant de pallier certains défauts de la μ -analyse, celle-ci peut maintenant être intégrée dans un processus itératif de synthèse et de validation de lois de commande non-linéaires. Basée sur les techniques d’inversion dynamique, une structure de correcteur constituée de deux boucles imbriquées est développée. La première boucle vise à linéariser partiellement le système et à rejeter ses principales non-linéarités. Le modèle linéaire obtenu est alors utilisé pour synthétiser la seconde boucle d’asservissement robuste par synthèse \mathcal{H}_∞ structurée. Après avoir calculé la LFR du système commandé, on évalue ensuite sa robustesse par μ -analyse et, dans le cas où la marge de robustesse n’est pas satisfaisante, un pire-cas est extrait. Ce pire-cas est ajouté au modèle linéaire de synthèse, afin de calculer un nouveau correcteur robuste par synthèse \mathcal{H}_∞ multi-modèles. Ce procédé est alors itéré jusqu’à ce qu’un correcteur stabilisant toutes les configurations soit obtenu. Cette technique a donné lieu à une publication [Lesprier et al. 2014], et à un article de revue en attente de décision pour le journal *Aerospace Science and Technology*.

Afin de valider la méthode développée précédemment, la suite de l’étude a consisté à l’appliquer à un modèle d’avion civil et à développer des lois de guidage, afin de simuler la phase complète d’atterrissage automatique d’un avion en présence d’incertitudes pour diverses configurations de vol.

Enfin, la généricité de l’approche proposée a été évaluée via une mise en œuvre sur un drone à voilure fixe dont les dimensions et le domaine de vol diffèrent considérablement de l’avion civil. Pour cela, un modèle représentatif d’un drone de type aile volante, dénommé "Avion Jaune" et disponible à l’ONERA, a été établi, puis la méthode de synthèse a été mise en œuvre et validée en simulations, dans le

but final d'être implantée sur le système réel. Cette étude a donné lieu également à une publication [Lesprier et al. 2015a].

ORGANISATION DU MANUSCRIT

Le manuscrit est organisé en trois parties permettant de distinguer l'analyse de robustesse des systèmes LTI (chapitres 1 et 2), la synthèse de lois de pilotage (chapitres 3 et 4), et les deux applications aéronautiques (chapitres 5, 6 et 7).

Le *premier chapitre* vise à exposer les principales notions théoriques sur la μ -analyse, allant des définitions aux considérations algorithmiques pour le calcul des bornes de μ . Il s'attarde particulièrement sur la caractérisation de la borne supérieure à l'aide de la formulation par matrices (D,G), dont les limites sont mises en évidence.

Le *deuxième chapitre* présente les premières contributions réalisées au cours de cette thèse sur la réduction du conservatisme et du temps de calcul de la borne supérieur de μ . Trois nouveaux algorithmes sont proposés. Des exemples, tirés d'une liste de benchmarks réalistes où le conservatisme de la borne supérieure de μ est important, permettent d'illustrer et de valider les méthodes.

Le *troisième chapitre* introduit les méthodes de commande non-linéaire par inversion dynamique robuste. Il présente donc les principaux éléments théoriques liés aux techniques de linéarisation entrée-sortie et à la détermination de la boucle externe par synthèse \mathcal{H}_∞ .

Le *quatrième chapitre* est dédié au développement d'une technique de synthèse de lois de commande non-linéaire robuste, prenant notamment en compte des incertitudes paramétriques. Ce chapitre constitue la clé de voute de ces travaux de thèse, en associant la synthèse de lois de pilotage aux techniques d'analyse de robustesse développées dans les chapitres 1 et 2. Il présente donc la contribution de ces travaux sur la synthèse de lois.

Le *cinquième chapitre* donne quelques notions de mécanique du vol au lecteur non familier de la dynamique des engins aéronautiques. Il présente les principales équations qui servent de base aux deux applications exposées dans les chapitres suivants.

Le *sixième chapitre* porte sur une première application aéronautique des mé-

thodes développées. Après avoir décrit le modèle physique d'un avion civil de transport de passagers, des lois de pilotage longitudinale et latérale sont calculées en utilisant la technique développée au chapitre 4. L'objectif est l'atterrissage automatique par suivi d'ILS. Les lois de guidage ainsi que les phases d'arrondi et de décrochage en présence de vent latéral sont exposées, et de nombreuses simulations permettent de valider la méthode complète.

Enfin, le *septième* chapitre présente une application à un drone à voilure fixe type aile volante, afin de démontrer la forte adaptabilité de la méthode de synthèse. En plus du développement des lois de pilotage, il présente également la modélisation du drone considéré, allant de la détermination des coefficients aérodynamiques à la vérification du modèle par des essais en vol en boucle ouverte. L'objectif final est l'implantation des lois sur le système réel.

CONSEILS DE LECTURE

Chaque chapitre se veut le plus indépendant possible, mais le lecteur doit être conscient que le chapitre 4 fait résolument appel aux résultats et connaissances exposés dans la première partie de ce manuscrit. Il est donc conseillé de parcourir le chapitre 1 pour acquérir les notions de base sur l'analyse de robustesse des systèmes LTI. La troisième partie du document, purement applicative, met en relief les atouts des méthodes développées aux parties 1 et 2.

Enfin, dans un souci de concision, seuls les résultats nécessaires à la compréhension des techniques développées sont rappelés, généralement au premier chapitre de chacune des trois parties. Les contributions personnelles sont quant à elles détaillées.

Première partie

**Analyse de robustesse des
systèmes LTI**

NOTIONS THÉORIQUES SUR L'ANALYSE DE ROBUSTESSE DES SYSTÈMES LTI

SOMMAIRE

1.1	MODÉLISATION LFT ET INCERTITUDES STRUCTURÉES	13
1.1.1	Quelques rappels sur la transformation linéaire fractionnaire	13
1.1.2	Théorème du faible gain et structure des incertitudes	17
1.2	VALEUR SINGULIÈRE STRUCTURÉE ET μ -ANALYSE	18
1.2.1	Introduction à la valeur singulière structurée	18
1.2.2	Considérations sur la borne inférieure de μ	19
1.2.3	Borne supérieure de μ : approche par matrices (D,G)	25
1.2.4	Borne supérieure de μ : autres approches	29
1.3	DU CONSERVATISME DE LA BORNE SUPÉRIEURE DE μ	35
	CONCLUSION	38

Résumé

L'ANALYSE de robustesse des systèmes LTI est un thème qui a suscité de nombreuses études au cours des dernières décennies. Afin d'évaluer la stabilité robuste d'un système commandé, en présence d'incertitudes de modélisation ou de dynamiques négligées, la μ -analyse s'avère être un outil puissant qui a su montrer ses preuves. Une étape préliminaire et générique consiste à représenter dans un premier temps le système bouclé par une LFR. Ce chapitre vise à rappeler quelques notions sur ce point, ainsi qu'à présenter un bref état de l'art sur la μ -analyse, en s'attardant notamment sur le calcul des bornes de la valeur singulière structurée et le conservatisme de la borne supérieure.

La μ -analyse, introduite par [Doyle 1982], est considérée aujourd'hui comme une technique mature pour l'analyse de robustesse d'une large classe de systèmes linéaires à temps invariant (ou LTI), en présence d'incertitudes paramétriques et de dynamiques négligées. Elle est suffisamment efficace pour être applicable sur des problématiques industrielles, ce qui accentue son intérêt dans notre étude.

Ce chapitre a pour objectif de présenter un état de l'art sur les notions clés liées à la μ -analyse. Tout d'abord la section 1.1 présente le formalisme à utiliser afin de pouvoir évaluer la stabilité robuste d'un système : la transformation linéaire fractionnaire, ou LFT. La valeur singulière structurée μ est ensuite introduite à la section 1.2. Elle permet de calculer théoriquement une marge de stabilité afin d'évaluer le domaine maximal admissible d'incertitudes avant que le système devienne instable. Cependant, le calcul de la valeur singulière structurée est un problème NP-difficile, ce qui pose des difficultés numériques en pratique. Il faut donc évaluer ses bornes inférieure et supérieure, et les principales méthodes de calcul de ces dernières sont rappelées au sein de cette même section. Enfin, la section 1.3 s'attarde sur un des problèmes inhérents à ce calcul de bornes : le conservatisme de la borne supérieure. Celle-ci est en effet souvent surévaluée, entraînant un écart trop important avec la borne inférieure, qui est quant à elle généralement plus proche de la valeur exacte de μ . Bien qu'une marge de stabilité soit obtenue, cet effet peut empêcher de conclure précisément sur la stabilité robuste d'un système. Des solutions à ce problème seront proposées dans le chapitre 2.

1.1 MODÉLISATION LFT ET INCERTITUDES STRUCTURÉES

1.1.1 Quelques rappels sur la transformation linéaire fractionnaire

Définition

Les LFTs (acronyme de l'anglais *Linear Fractional Transformations*) jouent un rôle important dans la théorie de la commande robuste. Ce sont des objets qui sont particulièrement utiles pour l'analyse de robustesse car ils permettent de mettre en évidence les incertitudes dynamiques et paramétriques. Une expression de type LFT entre un système augmenté et le correcteur est également introduite lors de la formulation du problème \mathcal{H}_∞ standard (voir chapitre 3), ce qui rend cette notion pertinente aussi en synthèse.

Pour reprendre ce qui est exposé dans [Zhou et al. 1996], un opérateur $F : \mathbb{C} \mapsto \mathbb{C}$, transformant une variable complexe $s \in \mathbb{C}$ en :

$$F(s) = \frac{a + bs}{c + ds} \quad (1.1)$$

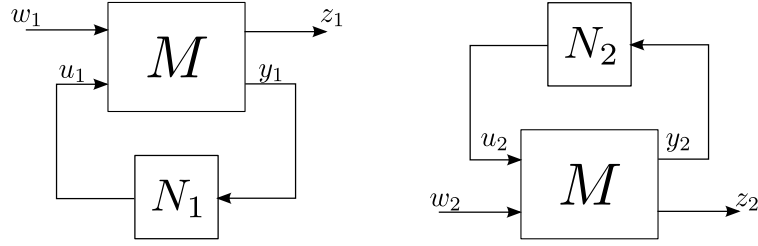


FIGURE 1.1 – LFTs inférieure (gauche) et supérieure (droite).

où a , b , c et d sont des scalaires complexes, est une transformation linéaire fractionnaire. En considérant $c \neq 0$, on peut écrire $F(s)$ de manière équivalente sous la forme :

$$F(s) = \alpha + \beta s(1 - \gamma s)^{-1} \quad (1.2)$$

où les scalaires complexes α , β et γ peuvent être exprimés en fonction de a , b , c et d . Il est possible de généraliser cette expression pour des matrices. Soit $M \in \mathbb{C}^{(n_1+n_2) \times (m_1+m_2)}$ la matrice complexe partitionnée comme suit :

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \quad (1.3)$$

et soient $N_1 \in \mathbb{C}^{m_2 \times n_2}$ et $N_2 \in \mathbb{C}^{m_1 \times n_1}$ deux autres matrices complexes. On peut définir deux opérateurs représentant les transformations linéaires fractionnaires inférieure et supérieure :

$$\mathcal{F}_l(M, \bullet) : \mathbb{C}^{m_2 \times n_2} \mapsto \mathbb{C}^{n_1 \times m_1} \quad (1.4)$$

$$\mathcal{F}_u(M, \bullet) : \mathbb{C}^{m_1 \times n_1} \mapsto \mathbb{C}^{n_2 \times m_2} \quad (1.5)$$

possédant les expressions suivantes :

$$\mathcal{F}_l(M, N_1) = M_{11} + M_{12}N_1(I - M_{22}N_1)^{-1}M_{21} \quad (1.6)$$

$$\mathcal{F}_u(M, N_2) = M_{22} + M_{21}N_2(I - M_{11}N_2)^{-1}M_{12} \quad (1.7)$$

et représentant respectivement les transferts entre w_1 et z_1 d'une part, et entre w_2 et z_2 d'autre part (voir Figure 1.1). Il est bien entendu supposé dans les expressions précédentes que les inverses existent, on parle alors de LFT *bien-posée*.

Définition 1.1

La LFT $\mathcal{F}_l(M, N_1)$ (resp. $\mathcal{F}_u(M, N_2)$) est dite bien-posée si $(I - M_{22}N_1)$ (resp. $(I - M_{11}N_2)$) est inversible.

On suppose désormais que toute LFT introduite est bien posée. L'expression utilisée dépend quant à elle du type de problème considéré. Dans le cadre de la com-

mande des systèmes ou de l'analyse de robustesse, il est convenu que la matrice M peut être remplacée par $M(s) \in \mathcal{RH}_\infty^{n \times n}$, un système linéaire temps invariant stable (i.e. possédant tous ses pôles dans le demi-plan complexe gauche), et les expressions précédentes restent bien entendu valides. Souvent, lorsqu'il s'agit d'analyse de robustesse on préfère la LFT supérieure, alors que pour la synthèse d'un correcteur on préfère la LFT inférieure. En ce qui concerne l'interprétation, par exemple pour la LFT supérieure $\mathcal{F}_u(M(s), N_2)$, on considère que le système nominal, représenté par $M_{22}(s)$, est perturbé par N_2 . Ainsi $M_{12}(s)$, $M_{21}(s)$ et $M_{11}(s)$ reflètent la connaissance de comment la perturbation N_2 affecte le système nominal $M_{22}(s)$. Du point de vue de l'analyse de robustesse, on remplace généralement la matrice N_2 par un opérateur $\Delta(s)$ dynamique, structuré ou non, de même dimensions que N_2 . On obtient alors une représentation où toute la dynamique fixée et connue est regroupée dans le système $M(s)$, alors que les incertitudes et les paramètres variants sont stockés au sein de cet opérateur $\Delta(s)$. Divers outils existent pour analyser la stabilité de tels systèmes, comme la μ -analyse introduite dans la section 1.2.

On s'intéresse à partir de maintenant, dans le cadre de l'analyse de robustesse de systèmes LTI, à la forme standard $M(s) - \Delta(s)$ représentée sur la figure 1.2, qu'on dénomme LFR (Représentation Fractionnaire Linéaire).

L'obtention d'une LFR n'est pas chose aisée, d'autant plus lorsque le système considéré contient un certain nombre d'incertitudes plus ou moins répétées. Il est important de comprendre que l'analyse de robustesse dépend très fortement de cette étape de modélisation. Une LFR qui possède une complexité trop importante peut être source de conservatisme ou bien poser des problèmes numériques tels qu'il n'est pas possible de trouver une solution au problème d'analyse.

De l'obtention d'une LFR

Il n'est pas question de détailler toutes les façons d'obtenir une LFR, mais plutôt de donner quelques pistes et outils pour son calcul numérique. Dans un contexte applicatif et méthodologique, on ne cherche pas en effet à évaluer une expression formelle pour tout type de problème rencontré. Le lecteur pourra trouver notamment dans Zhou et al. [1996] de nombreux détails à ce sujet.

On se place ici dans le contexte de l'analyse de robustesse en présence de diverses configurations paramétriques et incertitudes de modélisation. Il est important d'avoir un moyen numérique pour que, à partir d'une banque de données ou de modèles multivariés LTI, ou encore à partir de fonctions non-linéaires (issues par exemple d'équations représentant la dynamique d'un système complexe), une LFR puisse être générée. Pour ce faire, il existe deux étapes :

1. obtenir une formulation rationnelle (ou, à moindre niveau, polynomiale) en les paramètres du système considéré, puis :
2. transformer cette formulation rationnelle en LFR.

Etape 1 La première étape est souvent la plus critique, puisqu'elle est fortement dépendante du problème considéré et des données possédées. Il y a plusieurs possibilités :

- lorsqu'on possède des données tabulées (par exemple, des coefficients aérodynamiques ou des gains de correcteur dont les valeurs sont données pour diverses configurations de vol), il faut trouver une expression polynomiale ou rationnelle de ces coefficients en fonction de paramètres. Dans le cas polynomial, l'approche la plus efficace est l'approximation des moindres carrés linéaires surtout lorsqu'elle intègre un processus préliminaire d'orthogonalisation. Ce dernier permet de réduire sensiblement le nombre de monômes et donc la complexité de l'expression obtenue (voir par exemple [Döll et al. 2008; Roos 2009]). Dans le cas rationnel, on peut citer les moindres-carrés non-linéaires (une implémentation sur Matlab est disponible via la Curve Fitting Toolbox [The MathWorks 2015a]), la programmation quadratique (voir [Celis et al. 2007]), ou encore des méthodes plus récentes comme les algorithmes génétiques (voir [Hardier et al. 2013a]) ou les modèles de substitution (voir [Hardier et al. 2013b]). Toutes ces techniques sont disponibles dans la librairie APRICOT (*Approximation of Polynomial and Rational-type for Indeterminate Coefficients via Optimization Tools*), détaillée dans Roos et al. [2014], de la boîte à outils Matlab SMAC (*Systems Modeling Analysis and Control*) (voir SMAC [2012-2015]).
- dans le cas où l'on possède des fonctions non-linéaires analytiques de paramètres variants ou incertains (équations de la dynamique du vol par exemple), il existe deux méthodes.
 1. Une banque de modèles LTI est générée par linéarisations successives autour de différents points de fonctionnement, puis une approximation rationnelle est générée par les méthodes décrites ci-dessus (puisque l'on possède une série de données tabulées). La méthode proposée dans [Poussot-Vassal et Roos 2012] ajoute une surcouche afin de s'assurer que tous les modèles LTI obtenus sont bien exprimés dans une même base (modale ou compagne par exemple).
 2. Les fonctions non-linéaires sont remplacées par leurs approximations rationnelles (par exemple par développements limités à un certain ordre).

Etape 2 Une fois une expression polynomiale ou rationnelle obtenue, la seconde étape consiste à générer une LFR correspondante. Des outils puissants compatibles avec Matlab permettent de réaliser cette étape de façon automatique et efficace, même si aucune garantie de minimalité n'est assurée (voir notamment la LFR Tool-box de Magni [2006] et la documentation associée).

1.1.2 Théorème du faible gain et structure des incertitudes

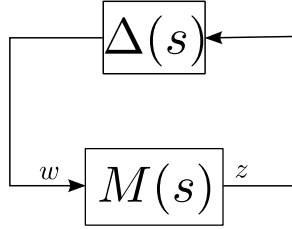


FIGURE 1.2 – Interconnexion standard pour l'analyse de stabilité robuste.

On considère l'interconnexion standard représentée sur la figure 1.2, où $M(s) \in \mathcal{RH}_\infty^{p \times p}$ désigne un système LTI stable, modélisant par exemple un système bouclé nominal. $\Delta(s) \in \mathcal{RH}_\infty^{p \times p}$ est un opérateur borné en gain mais de phase aléatoire, qui représente l'écart entre le modèle $M(s)$ et le système réel. Il est pour le moment non-structuré. Le théorème suivant fournit une condition nécessaire et suffisante de stabilité robuste dans ce cas.

Théorème 1.1 (Théorème du faible gain, version non structurée)

Soit $M(s)$ un système LTI stable. L'interconnexion de la figure 1.2 est stable pour toute incertitude LTI stable $\Delta(s)$ telle que $\|\Delta(s)\|_\infty \leq 1$ si et seulement si $\|M(s)\|_\infty \leq 1$.

Le problème de cette caractérisation est qu'elle peut s'avérer très conservatrice. En effet, les incertitudes qui affectent le système $M(s)$ correspondent généralement à des paramètres physiques. Elles agissent en des points bien précis et de façon très structurée. Avoir $\Delta(s)$ non-structuré revient à considérer une incertitude dynamique globale uniquement bornée en gain, ce qui est donc peu représentatif du système réel. La classe des systèmes incertains considérée dans ce cas est donc beaucoup trop grande.

Il est par conséquent important de structurer $\Delta(s)$ en le remplaçant par un opérateur bloc-diagonal :

$$\Delta(s) = \text{diag}(\delta_1 I_{k_1}, \dots, \delta_r I_{k_r}, \Delta_1(s), \dots, \Delta_q(s)) \quad (1.8)$$

où les δ_i peuvent être des scalaires réels ou complexes (éventuellement répétés si $k_i > 1$), alors que les $\Delta_j(s)$ correspondent généralement à des dynamiques négligées, sous la forme de fonctions de transfert stables non structurées. Sans perte

de généralité, les δ_i seront toujours considérés comme réels, un scalaire complexe n'ayant en effet aucun sens physique et présentant donc peu d'intérêt dans ces travaux. Il faut cependant noter qu'il serait aisé de les considérer si besoin. On introduit alors la structure Δ :

$$\Delta = \left\{ \text{diag} (\delta_1 I_{k_1}, \dots, \delta_r I_{k_r}, \Delta_1, \dots, \Delta_q), \delta_i \in \mathbb{R}, \Delta_j \in \mathbb{C}^{l_j \times l_j} \right\} \quad (1.9)$$

avec $p = \sum_{i=1}^r k_i + \sum_{j=1}^q l_j$, où p est tel que $\Delta(j\omega) \in \Delta \subset \mathbb{C}^{p \times p}, \forall \omega \in \mathbb{R}$. Enfin on note $\mathcal{B}_\Delta = \{\Delta \in \Delta : \bar{\sigma}(\Delta) \leq 1\}$ et plus généralement $k\mathcal{B}_\Delta = \{\Delta \in \Delta : \bar{\sigma}(\Delta) \leq k\}$, où $\bar{\sigma}(\cdot)$ représente la plus grande valeur singulière.

1.2 VALEUR SINGULIÈRE STRUCTURÉE ET μ -ANALYSE

1.2.1 Introduction à la valeur singulière structurée

Après avoir défini une structure d'incertitudes pour modéliser au mieux les systèmes LTI incertains, la notion de valeur singulière structurée, dénotée μ_Δ , est introduite dans cette section. C'est le fondement de la μ -analyse, qui cherche à déterminer la marge de robustesse, dénotée k_{max} . Cette dernière correspond à la plus grande valeur de k telle que l'interconnexion de la figure 1.2 est stable pour toute incertitude structurée $\Delta \in k\mathcal{B}_\Delta$.

Il est facile d'observer que l'interconnexion de la figure 1.2 devient instable lorsque $\det(I - \Delta(s)M(s)) = 0$, pour un certain s imaginaire pur. Afin d'introduire la définition de la valeur singulière structurée, on se place à une pulsation ω et on note $M = M(j\omega) \in \mathbb{C}^{n \times n}$ la réponse fréquentielle de $M(s)$ à cette fréquence. On cherche alors la plus petite valeur k pour laquelle il existe une incertitude $\Delta \in k\mathcal{B}_\Delta$ telle que $\det(I - \Delta M) = 0$. Dans le cas où Δ n'est pas structurée ($\Delta = \mathbb{C}^{p \times p}$), on montre que k correspond à l'inverse de la plus grande valeur singulière, soit $k = \bar{\sigma}(M)^{-1}$. On se ramène alors bien au cas du théorème 1.1 du faible gain. Pour le cas où $\Delta \in \Delta$ avec Δ défini selon l'équation (1.9), on remplace la plus grande valeur singulière par la notion de valeur singulière structurée, qui a été introduite par Doyle [1982].

Définition 1.2 (Valeur singulière structurée)

Soit $M = M(j\omega)$ la réponse fréquentielle du système LTI stable $M(s)$ à une fréquence ω . Si aucune matrice $\Delta \in \Delta$ ne rend la matrice $I - \Delta M$ singulière, alors la valeur singulière structurée est définie par $\mu_\Delta(M) = 0$. Sinon :

$$\mu_\Delta(M) = \frac{1}{\min \{k \in \mathbb{R}^+ : \exists \Delta \in k\mathcal{B}_\Delta, \det(I - \Delta M) = 0\}} \quad (1.10)$$

L'inverse de la valeur singulière structurée donne donc, à une fréquence, la taille de la plus petite perturbation structurée Δ telle que l'interconnexion de la figure 1.2 est en limite de stabilité. En généralisant ceci à tout le continuum fréquentiel, la marge de robustesse correspond à l'inverse de la plus grande valeur singulière structurée évaluée à toutes les fréquences réelles positives :

Définition 1.3 (Marge de robustesse)

L'inverse de la valeur maximale de $\mu_\Delta(M(j\omega))$ sur la plage de fréquence \mathbb{R}^+ donne la marge de robustesse k_{max} :

$$k_{max} = \left[\sup_{\omega \in \mathbb{R}^+} \mu_\Delta(M(j\omega)) \right]^{-1} \quad (1.11)$$

Ceci permet d'introduire la version structurée du théorème du faible gain, dont la marge de robustesse définie précédemment découle tout naturellement.

Théorème 1.2 (Théorème du faible gain, version structurée)

Soit $M(s)$ un système LTI stable. L'interconnexion de la Figure 1.2 est stable pour toute incertitude LTI stable $\Delta(s) \in \mathcal{B}_\Delta$ si et seulement si :

$$\max_{\omega \in \mathbb{R}} \mu_\Delta(M(j\omega)) \leq 1 \quad (1.12)$$

Il existe toutefois un problème majeur avec la valeur singulière structurée μ_Δ . Son calcul exact est NP-difficile dans le cas général (voir Braatz et al. [1994]). Cependant des algorithmes à temps polynomial ou exponentiel permettent d'évaluer des bornes inférieures et supérieures pour encadrer la valeur exacte. Les prochaines sections visent à donner quelques notions sur le calcul de ces bornes, utiles aux travaux qui suivront au chapitre 2.

1.2.2 Considérations sur la borne inférieure de μ

Le calcul d'une borne inférieure de μ_Δ pour toutes les fréquences $\omega \in \mathbb{R}^+$ permet de déterminer une configuration déstabilisante du système d'une part, et

surtout d'évaluer l'écart avec la borne supérieure d'autre part. Plus l'écart entre les deux bornes est faible, plus la connaissance de la vraie valeur de μ_Δ est précise. Un écart trop important est généralement problématique, signifiant que la marge de robustesse calculée est bien trop pessimiste. On reviendra sur cette question lors du chapitre 2.

De nombreux algorithmes existent afin de calculer une borne inférieure de μ_Δ . Basée sur les résultats de Fabrizio et al. [2014], une liste des principales méthodes est proposée ci-après. Pour la plupart (sauf la technique de migration de pôles), la fréquence est fixée durant l'optimisation, signifiant ainsi que $M(j\omega)$ et $\Delta(j\omega)$ sont des matrices constantes, qui seront notées M et Δ . Il conviendra donc d'utiliser un maillage fréquentiel. Il n'y a pas de risques, contrairement à l'évaluation de la borne supérieure de μ_Δ , de manquer une fréquence critique pour la marge de stabilité, puisqu'une borne inférieure obtenue sur un maillage sera forcément une borne inférieure sur \mathbb{R}^+ tout entier. Le seul risque concerne une évaluation trop conservative si le maillage est mal choisi.

Méthodes de complexité exponentielle

Les premières méthodes qui ont été développées pour le calcul d'une borne inférieure de $\mu_\Delta(M)$ sont à temps exponentiel. Une des plus connues, dans le cas d'incertitudes réelles non-répétées ($k_i = 1$ et $q = 0$ dans (1.9)), est celle de [De Gaston et Safonov 1988; Sideris et Sanchez Pena 1990]. Elle utilise le *mapping theorem* [Zadeh et Desoer 1963], qui montre que l'image de $k\mathcal{B}_\Delta$ par l'opérateur $\Delta \mapsto \det(I - \Delta M)$ est incluse dans l'enveloppe convexe des images des 2^r sommets de $k\mathcal{B}_\Delta$. On peut alors déterminer une borne inférieure de $\mu_\Delta(M)$ en étudiant la position de l'image de ces sommets par rapport à l'origine du plan complexe. Des extensions ont été proposées par [Sideris et De Gaston 1986; Sideris et Sanchez Pena 1989] pour le cas d'incertitudes réelles répétées. Cependant la limitation majeure de cette approche est le calcul des 2^r images des sommets de $k\mathcal{B}_\Delta$, qui peut entraîner un temps de calcul prohibitif dans le cas d'un grand nombre d'incertitudes.

Pour le même cas d'incertitudes réelles non-répétées, une méthode algébrique est proposée par [Dailey 1990]. Elle consiste à rechercher des incertitudes déstabilisantes $\Delta \in k\mathcal{B}_\Delta$ dont toutes les composantes δ_i sauf deux atteignent la valeur maximale k . Elle utilise des manipulations algébriques matricielles simples, et une dichotomie est réalisée pour déterminer la plus petite valeur de k . Il y a $r(r-1)/2$ façons de choisir 2 incertitudes parmi les r , et 2^{r-2} façons de fixer les $r-2$ autres à $-k$ ou k . Pour chaque valeur de k , il y a donc $r(r-1)2^{r-3}$ recherches qui sont exécutées, ce qui montre la complexité exponentielle de cette approche.

Enfin, une méthode pour le cas d'incertitudes réelles éventuellement répétées

($k_i \geq 1$) est proposée dans [Matsuda et al. 2009]. Elle introduit $b = \prod_{i=1}^r (k_i + 1) - 1$ incertitudes fictives $\bar{\delta}_1, \dots, \bar{\delta}_b$ qui sont exprimées en fonction de $\delta_1, \dots, \delta_r$, et réécrit $\det(I - M(s)\Delta)$ comme une fonction $f(s) = \hat{p}(s) + [\bar{\delta}_1 \dots \bar{\delta}_b][p_1(s) \dots p_b(s)]^T$, où $\hat{p}(s), p_1(s), \dots, p_b(s)$ sont des polynômes réels fixés. La stabilité de $f(s)$ est ensuite analysée grâce à l'outil dénommé *Stability Feeler* de [Matsuda et Mori 2009], permettant ainsi d'obtenir des bornes supérieure et inférieure de μ .

Il est évident que ces méthodes doivent être utilisées uniquement dans le cas d'un nombre limité d'incertitudes pour ne pas obtenir des temps de calcul trop prohibitifs, ce qui limite leur application. Par la suite, nous nous intéresserons donc uniquement aux méthodes à temps polynomial afin de pouvoir les utiliser dans le cas d'applications réalistes.

Power algorithm

Introduit par [Packard et al. 1988] dans le cas purement complexe, et adapté dans le cas des incertitudes mixtes réelles/complexes par [Young et Doyle 1990; 1997], cette technique vise à résoudre le problème d'optimisation non-concave :

$$\mu_{\Delta}(M) = \max_{Q \in \mathcal{Q}} \rho_R(QM) \quad (1.13)$$

où $\rho_R(M)$ désigne la plus grande valeur propre réelle de M en valeur absolue, et $\mathcal{Q} = \{Q \in \Delta : \delta_i \in [-1, 1], \Delta_j^* \Delta_j = I_{l_j}\}$. Le problème n'étant pas concave, il n'est pas possible de déterminer un maximum global (qui serait alors la valeur exacte de la valeur singulière structurée), mais seulement des maxima locaux. Le calcul d'un de ces maxima peut être exécuté en utilisant un algorithme du point fixe, ce qui conduit à une borne inférieure de $\mu_{\Delta}(M)$. Cette approche fonctionne bien dans le cas purement complexe ou mixte réel/complexe, et des améliorations pour le cas purement réel ont été proposées dans [Tierno et Young 1992; Newlin et Glavaski 1995]. Cependant, dans ce dernier cas des problèmes de convergence apparaissent fréquemment.

Ajout d'une petite quantité d'incertitudes complexes

Lorsque les incertitudes présentes dans le bloc Δ sont purement réelles, il arrive que la fonction $\omega \mapsto \mu_{\Delta}(M(j\omega))$ ne soit plus continue. Ceci peut conduire à une forte surestimation de la marge de robustesse en cas de résolution sur un maillage fréquentiel. Ce risque est bien plus faible dans le cas des incertitudes complexes ou mixtes du fait de la disparition de ce problème de discontinuité. L'idée, proposée par [Packard et Pandey 1993], est donc d'ajouter une petite quantité $\epsilon > 0$ d'incer-

titudes complexes au problème purement réel, de telle sorte que l'interconnexion $M - \Delta$ de la figure 1.2 soit remplacée par une interconnexion $M_\epsilon - \Delta_a$, avec :

$$M_\epsilon = \begin{bmatrix} M & \sqrt{\epsilon}M \\ \sqrt{\epsilon}M & \epsilon M \end{bmatrix} \quad (1.14)$$

et $\Delta_a = \text{diag}(\Delta, \Delta_c)$, où $\Delta \in \mathbf{\Delta}$ et Δ_c est une matrice complexe de la même taille et de la même structure que Δ . On montre alors que $\mu_{\Delta_a}(M_\epsilon) \geq \mu_{\Delta}(M)$ pour tout $\epsilon > 0$, et que $\mu_{\Delta_a}(M_\epsilon) \xrightarrow{\epsilon \rightarrow 0} \mu_{\Delta}(M)$.

Bien que donnant parfois de bons résultats, la principale difficulté vient du choix de ϵ . Pris trop grand, il change drastiquement le problème initial, et la marge de robustesse peut être très sous-estimée. Pris trop petit, le calcul d'une borne inférieure de $\mu_{\Delta_a}(M_\epsilon)$ peut être délicat numériquement. Un autre problème concerne aussi la taille du problème, qui est tout de même multiplié par deux. Même si le *power algorithm* présente un temps de calcul relativement rapide, ce dernier peut augmenter considérablement avec la nouvelle formulation.

Gain-based algorithm

Dans [Seiler et al. 2010], le calcul de $\mu_{\Delta}(M)$ est reformulé comme un problème de performance \mathcal{H}_∞ pire-cas. Pour les incertitudes réelles, l'algorithme proposé dans [Packard et al. 2000] est utilisé pour déterminer une suite de perturbations Δ faisant tendre la norme \mathcal{H}_∞ d'un certain transfert à l'infini, ce qui traduit l'apparition d'une instabilité dans l'interconnexion de la figure 1.2. Le *power algorithm* est quant à lui appliqué sur les incertitudes complexes.

Migration de pôles vers l'axe imaginaire

La migration de pôles a été proposée par [Magni et Döll 1997; Magni et al. 1999]. Elle est principalement dédiée aux problèmes purement réels – le cas complexe amenant généralement des problèmes de formulation du problème d'optimisation, ou de temps de calcul. Contrairement aux autres techniques, elle ne cherche pas à déterminer un profil de borne inférieure en fonction de la fréquence, mais à identifier directement des perturbations déstabilisantes dont la norme est la plus petite possible, et les fréquences qui y sont associées. Son principe est relativement intuitif, puisque l'idée consiste à déplacer une valeur propre de l'interconnexion entre $M(s)$ et Δ vers l'axe imaginaire. En considérant les matrices de la représentation d'état de $M(s)$, notées (A, B, C, D) , la matrice d'état de l'interconnexion $M(s) - \Delta$ est donnée par $A_0 = A + B\Delta(I - D\Delta)^{-1}C$. La marge de robustesse définie en (1.11)

peut alors être donnée de façon équivalente par :

$$k_{max} = \min_{\Delta \in \Delta} \{ \bar{\sigma}(\Delta), \lambda_{max}(A_0) = 0 \} \quad (1.15)$$

où $\lambda_{max}(A_0)$ représente la plus grande partie réelle des valeurs propres de A_0 .

Une première approche consiste à utiliser une caractérisation du premier ordre de la variation des pôles du système incertain causée par une petite variation $d\Delta$ de Δ . Le calcul d'une série de plusieurs petites perturbations $d\Delta$ permet de déplacer progressivement les pôles vers l'axe imaginaire. Il existe deux algorithmes pour résoudre ce problème.

- Le premier consiste à déterminer une perturbation de norme de Frobenius la plus petite possible conduisant un pôle sur l'axe imaginaire. Ensuite, une perturbation de norme infinie la plus petite possible qui maintient le système en limite de stabilité est déterminée. Du fait que cet algorithme ne nécessite pas de balayage en fréquences, il est beaucoup plus rapide que ceux exposés précédemment.
- Le deuxième algorithme vise à utiliser le *power algorithm* sur une interconnexion régularisée afin de ne pas subir les problèmes de convergence dans le cas purement réel. Ainsi trois étapes sont proposées. La première consiste à ajouter une petite quantité d'incertitudes complexes comme expliqué précédemment. La deuxième applique le *power algorithm* sur un maillage fréquentiel grossier au problème ainsi régularisé. Un pôle du système initial est généralement amené à proximité de l'axe imaginaire une fois cette étape réalisée. La perturbation déstabilisante obtenue sert ensuite d'initialisation à la dernière étape, qui résout un problème de programmation linéaire afin de faire migrer ce pôle sur l'axe imaginaire. La valeur $\hat{\Delta}$ de Δ obtenue est la perturbation de norme $\bar{\sigma}$ minimum qui permet de déstabiliser une valeur propre de l'interconnexion. La borne inférieure est alors donnée par $\bar{\sigma}(\hat{\Delta})^{-1}$. Cette méthode est globalement très efficace et rapide.

Enfin et plus récemment, une approche directe a été proposée par [Iordanov et al. 2003; Iordanov et Halton 2014]. Elle considère le problème d'optimisation :

$$\min_{\Delta \in \Delta} \bar{\sigma}(\Delta) \text{ tel que } \lambda_{max}(A_0) = 0 \quad (1.16)$$

qui peut être reformulé comme un problème de programmation quadratique séquentielle et résolu à l'aide de la boîte à outils Matlab Optimization Toolbox [The MathWorks 2015b]. Afin d'éviter des problèmes de convergence, la condition $\lambda_{max}(A_0) = 0$ peut être remplacée par $\lambda_{max}(A_0) \leq \epsilon$ avec un ϵ défini assez petit par l'utilisateur. Il faut enfin noter qu'étant donné que le problème (1.16) est non

convexe, les résultats dépendent grandement du choix initial de Δ . Cette technique est assez efficace mais relativement lente (l'algorithme utilisé est plus lent qu'un algorithme de programmation linéaire).

Techniques d'optimisation directe

Plusieurs méthodes utilisent directement l'expression classique de la valeur singulière structurée présentée dans la définition 1.2. Elles sont bien adaptées au cas d'incertitudes purement réelles. Par contre, les temps de calcul augmentent de manière significative en présence d'incertitudes complexes.

La première consiste à résoudre le problème non-convexe suivant :

$$\min_{\Delta \in \Delta} \bar{\sigma}(\Delta) \text{ tel que } \det(I - M\Delta) = 0 \quad (1.17)$$

via l'utilisation d'outils d'optimisation non-linéaire comme la routine `fmincon` de la Matlab Optimization Toolbox [The MathWorks 2015b]. Afin d'éviter des problèmes de convergence, il est proposé dans [Hayes et al. 2001; Bates et Mannchen 2004] de remplacer la contrainte égalité par $\underline{\sigma}(I - M\Delta) \leq \epsilon$, ou dans [Halton et al. 2008] par $|\det(I - M\Delta)| \leq \epsilon$, où ϵ est une petite tolérance définie par l'utilisateur.

Une autre formulation est donnée dans [Yazıcı et al. 2011] dans le cas d'incertitudes purement réelles éventuellement répétées :

$$\min_{\substack{\Delta \in \Delta \\ \lambda_1, \lambda_2 \in \mathbb{R}}} \bar{\sigma}(\Delta) + (\lambda_1^q + \lambda_2^q)\lambda \text{ tel que } \begin{cases} \operatorname{Re}(\det(I - M\Delta)) = \lambda_1^p \\ \operatorname{Im}(\det(I - M\Delta)) = \lambda_2^p \end{cases} \quad (1.18)$$

où p et q sont des entiers impair et pair respectivement, et $\lambda \geq 10^5$ est une pénalité suffisamment large. Les scalaires réels λ_1 et λ_2 permettent à la fois de contribuer positivement à la fonction objectif avec leurs puissances paires, et de relâcher les contraintes selon des directions négatives ou positives du fait des puissances impaires. Ce problème peut être résolu par un algorithme modifié de descente de gradients introduit par [Kasimbeyli et al. 2009].

[Brito et Kim 2010] propose une variation qui consiste dans un premier temps à résoudre le problème suivant pour différentes valeurs de k et pour un petit ϵ prédéfini :

$$g(k) = \min_{\Delta \in k\mathcal{B}_\Delta} \operatorname{Re}(\det(I - M\Delta)) \text{ tel que } |\operatorname{Im}(\det(I - M\Delta))| \leq \epsilon \quad (1.19)$$

en utilisant la fonction `fmincon` de la Matlab Optimization Toolbox. Une variante de la méthode de Newton-Raphson est ensuite appliquée afin de déterminer la plus

petite valeur \underline{k} de k telle que $g(k) = 0$. L'inverse de \underline{k} donne une bonne inférieure de $\mu_{\Delta}(M)$.

Enfin, il est à noter que l'objectif est de minimiser la fonction non-lisse (1.17), et il n'existe aucune garantie que l'utilisation d'outils d'optimisation non-linéaire classiques comme proposé ci-avant permette d'obtenir un minimum local. [Lemos et al. 2014] propose donc d'utiliser des techniques spécifiques d'optimisation non-lisse qui assurent la convergence vers un minimum local de la résolution de (1.17).

Exploitation de propriétés géométriques

Dans le cas d'incertitudes réelles éventuellement répétées, une approche géométrique est proposée dans [Kim et al. 2008]. Elle combine tirages aléatoires et optimisation. Elle consiste à calculer les signes des parties réelle et imaginaire de $\det(I - \Delta M)$ pour différentes incertitudes Δ choisies aléatoirement sur la frontière d'un hypercube donné contenu dans l'espace d'incertitudes \mathbb{R}^r . L'hypercube est agrandi jusqu'à ce que les quatre combinaisons de signes soient rencontrées, à savoir $(-1, -1)$, $(-1, +1)$, $(+1, -1)$ et $(+1, +1)$, ce qui signifie qu'il peut contenir des valeurs $\delta_1, \dots, \delta_r$ telles que $\det(I - \Delta M) = 0$. Des contractions et expansions sont ensuite exécutées jusqu'à ce que la taille de l'hypercube contenant une perturbation déstabilisante devienne plus petite qu'une tolérance prédéfinie.

La méthode repose principalement sur la probabilité de détecter une incertitude déstabilisante, qui elle-même dépend du nombre de tirages réalisés. Le temps de calcul de la technique est ainsi très dépendant de ce nombre de tirages, et est une fonction polynomiale du nombre d'incertitudes réelles. Il est cependant indépendant du nombre de répétitions de ces dernières.

Toutes ces méthodes de calcul de borne inférieure de μ_{Δ} permettent de traiter un bon nombre de systèmes et d'obtenir une borne satisfaisante dans la majorité des cas : par exemple, le *power algorithm* donne d'excellents résultats dans le cas purement complexe ou mixte, alors que la migration de pôles est la plus efficace dans le cas purement réel. Il sera indiqué plus en détails, à la section 1.3 traitant du conservatisme des bornes de μ_{Δ} , quelles sont les méthodes qui fonctionnent le mieux suivant le type d'incertitudes rencontré.

1.2.3 Borne supérieure de μ : approche par matrices (D,G)

Cette sous-section vise à présenter la méthode la plus communément utilisée afin de calculer une borne supérieure de μ . Elle est en effet à temps polynomial ce qui en fait une méthode particulièrement intéressante pour des applications

réalistes. Il existe de plus des méthodes efficaces permettant de l'évaluer sur l'ensemble du domaine fréquentiel considéré sans l'utilisation d'un maillage. On reviendra cependant sur son principal défaut – le conservatisme – à la fin de ce chapitre. Des alternatives à temps exponentiel seront également exposées à la sous-section suivante.

Calcul de la borne supérieure à une fréquence

L'approche la plus classique et la plus utilisée pour le calcul d'une borne supérieure de μ à une fréquence donnée nécessite d'introduire deux ensembles de matrices de scaling :

$$\begin{aligned}\mathcal{D}_n &= \{D \in \mathbb{C}^{n \times n}, D = D^* > 0 : \forall \Delta \in \Delta, D\Delta = \Delta D\} \\ \mathcal{G}_n &= \{G \in \mathbb{C}^{n \times n}, G = G^* : \forall \Delta \in \Delta, G\Delta = \Delta^* G\}\end{aligned}\quad (1.20)$$

La borne associée, initialement proposée par [Fan et al. 1991] et reprise par [Young et al. 1995], est alors rappelée dans la proposition suivante :

Proposition 1.1

Soit β un scalaire positif. Soient $M = M(j\omega)$ la réponse fréquentielle de $M(s)$ à une pulsation ω , et Δ une structure donnée. S'il existe des matrices $D \in \mathcal{D}_n$ et $G \in \mathcal{G}_n$ telles que :

$$M^*DM + j(GM - M^*G) \leq \beta^2 D \quad (1.21)$$

ou, de façon équivalente, avec $\tilde{D} = D^{1/2}$ et $\tilde{G} = \beta^{-1}D^{-1/2}GD^{-1/2}$:

$$\bar{\sigma} \left(\left(\frac{\tilde{D}M\tilde{D}^{-1}}{\beta} - j\tilde{G} \right) (I + \tilde{G}^2)^{-\frac{1}{2}} \right) \leq 1 \quad (1.22)$$

ou encore, avec $\hat{D} = P$ et $\hat{G} = V^*\tilde{G}V$, où P est une matrice hermitienne définie positive et V une matrice unitaire telles que $\tilde{D} = VP$:

$$\bar{\sigma} \left((I + \hat{G}^2)^{-\frac{1}{4}} \left(\frac{\hat{D}M\hat{D}^{-1}}{\beta} - j\hat{G} \right) (I + \hat{G}^2)^{-\frac{1}{4}} \right) \leq 1 \quad (1.23)$$

alors $\mu_\Delta(M) \leq \beta$.

La minimisation de β dans la précédente proposition donne donc une borne supérieure de $\mu_\Delta(M)$, que l'on dénotera $\bar{\mu}_\Delta(M)$ dans la suite. Afin de calculer cette borne, on utilise généralement un solveur LMI via la caractérisation (1.21) [Beck et Doyle 1992], ou un algorithme de descente de gradient en utilisant les caractérisations (1.21)-(1.23) [Young et al. 1992; 1995]. Cette dernière approche se révèle très intéressante du point de vue de la complexité numérique, mais est sous-optimale par rapport à la résolution de LMIs. Cette sous-optimalité amène donc

dans certains cas à une marge de robustesse trop pessimiste. Ce problème sera traité plus en détails lors du chapitre 2.

Il existe également d'autres méthodes pour calculer $\bar{\mu}_\Delta(M)$:

- [Fransson 2004] propose d'utiliser une stratégie d'optimisation à deux niveaux. Le principe consiste à déterminer des valeurs D_{opt} et G_{opt} solutions de (1.21) par SDP (*Semi-Definite Programming*), et ensuite à utiliser une technique de dichotomie sur β , en exploitant le fait que la fonction définie à partir de (1.21) par $V(\beta, D_{opt}, G_{opt}) = \beta^2 D_{opt} - M^* D_{opt} M - j(G_{opt} M - M^* G_{opt})$ est singulière pour la valeur optimale β_{opt} que l'on cherche à déterminer. Cette méthode est simple à mettre en œuvre mais peut amener un temps de calcul plus long que les techniques classiques.
- Une alternative au solveur LMI qui cherche à résoudre un problème aux valeurs propres généralisé (routine `gevp` sur Matlab), consiste à utiliser la méthode de Newton avec une technique de dichotomie [Chen et al. 2010], ce qui permet d'améliorer le temps de calcul.
- Enfin, une extension de la formulation (D,G) est proposée dans [Scorletti et al. 2007]. Une autre matrice de scaling notée L est introduite afin de mieux prendre en compte la structure réelle des incertitudes, tout en gardant une complexité polynomiale de résolution. Ceci permet de réduire le conservatisme de la formulation classique, notamment dans le cas où la matrice Δ possède un très grand nombre d'incertitudes réelles répétées.

Evaluation de $\bar{\mu}_\Delta(M(j\omega))$ pour toute pulsation $\omega \in \mathbb{R}^+$

La proposition 1.1 permet seulement de calculer $\bar{\mu}_\Delta(M)$ à une fréquence donnée. Le calcul de $\bar{\mu}_\Delta(M(j\omega))$ pour toute fréquence $\omega \in \mathbb{R}^+$ est cependant nécessaire afin d'obtenir une borne inférieure de la marge de robustesse k_{max} , et ainsi une garantie (pessimiste) de stabilité robuste. Il s'agit d'un problème de dimension infinie qui ne peut être résolu en l'état.

Une solution consiste à utiliser un maillage fréquentiel, mais cette méthode présente de nombreux inconvénients. Le premier est qu'aucune garantie ne peut être donnée quant à la validité de la marge de robustesse obtenue. En effet, rien n'assure qu'une fréquence critique n'est pas laissée de côté, correspondant à la valeur maximale de $\mu_\Delta(M(j\omega))$. Dans le cas des systèmes flexibles notamment, la fonction $\omega \mapsto \mu_\Delta(M(j\omega))$ peut être discontinue et posséder des pics très élevés et très étroits. On risque donc d'évaluer une marge de robustesse trop optimiste, voire de manquer certains cas critiques pouvant révéler une instabilité. De plus, l'utilisation d'un maillage fin amène des temps de calcul prohibitifs lorsqu'un grand nombre de fréquences est pris en considération. Cette méthode du maillage fréquentiel n'est

donc pas envisageable en pratique. Afin de pallier les défauts précédents, il existe principalement deux stratégies.

La première consiste à considérer la fréquence comme une incertitude réelle répétée. On introduit donc un bloc d'incertitudes augmenté $\tilde{\Delta}(s) = \text{diag}(\omega I_n, \Delta(s))$, où n représente l'ordre du système [Sideris 1992]. Une telle approche dépend fortement de l'ordre du système considéré. En effet, si n est petit, elle fonctionne assez bien, mais si n est grand, alors des problèmes de conservatisme peuvent se présenter tout comme un temps de calcul élevé du fait du grand nombre de variables d'optimisation dans les matrices (D, G) associées au bloc ωI_n . La résolution de ce problème augmenté nécessite d'introduire la valeur singulière structurée dissymétrique [Ferrerres et Fromion 1997; Holland et al. 2005; Halton et al. 2008] et fournit une marge de robustesse sur une plage de fréquences donnée.

La seconde stratégie est basée sur des techniques de balayage fréquentiel, notamment présentées dans [Biannic et Ferrerres 2005; Roos et Biannic 2010]. Ces dernières permettent de déterminer les intervalles de fréquences pour lesquels β et les matrices (D, G) , évalués par la résolution de (1.21), (1.22), ou (1.23) restent valables. Cette stratégie est ensuite répétée pour un certain nombre de fréquences, jusqu'à ce que l'union de tous les intervalles soit égale à \mathbb{R}^+ . Ceci permet d'obtenir une borne supérieure de μ_Δ sur l'ensemble du continuum fréquentiel, sans le risque de manquer une fréquence critique, et avec un temps de calcul relativement faible (car le nombre de fréquences à considérer est généralement peu élevé).

On rappelle ici la méthode pour déterminer l'ensemble des fréquences pour lesquelles β , et les matrices de scaling restent valables. On considère le triplet $(\beta_i, \hat{D}(\omega_i), \hat{G}(\omega_i))$ qui vérifie l'inégalité (1.23) pour une fréquence donnée ω_i avec $M = M(j\omega_i)$. Le but est de déterminer le plus grand intervalle $I(\omega_i) \ni \omega_i$ pour lequel le triplet précédent reste valable, i.e. tel que $\forall \omega \in I(\omega_i)$:

$$\bar{\sigma} \left(F(\omega_i)^{-\frac{1}{4}} \left(\frac{\hat{D}(\omega_i) M(j\omega) \hat{D}(\omega_i)^{-1}}{\beta_i} - j\hat{G}(\omega_i) \right) F(\omega_i)^{-\frac{1}{4}} \right) \leq 1 \quad (1.24)$$

avec $F(\omega_i) = I + \hat{G}(\omega_i)^2$. On a alors la proposition suivante, extraite de [Roos et Biannic 2010].

Proposition 1.2

Soit (A_M, B_M, C_M, D_M) une représentation d'état de $M(s)$. On pose alors la matrice Hamiltonienne suivante :

$$\mathcal{H} = \begin{bmatrix} A_H & 0 \\ -C_H^* C_H & -A_H \end{bmatrix} + \begin{bmatrix} B_H \\ -C_H^* D_H \end{bmatrix} X \begin{bmatrix} D_H^* C_H & B_H^* \end{bmatrix} \quad (1.25)$$

où $X = (I - D_H^* D_H)^{-1}$ et :

$$\begin{bmatrix} A_H & B_H \\ C_H & D_H \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & \frac{F^{-\frac{1}{4}}}{\sqrt{\beta_i}} \end{bmatrix} \begin{bmatrix} A_M - j\omega_i I & B_M \hat{D}^{-1} \\ \hat{D} C_M & \hat{D} D_M \hat{D}^{-1} - j\beta_i \hat{G} \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \frac{F^{-\frac{1}{4}}}{\sqrt{\beta_i}} \end{bmatrix} \quad (1.26)$$

où la dépendance en ω_i a été omise par soucis de clarté. Soit δ_{ω_i-} et δ_{ω_i+} tels que :

$$\begin{aligned} \delta_{\omega_i-} &= \max\{\lambda \in \mathbb{R}^- : \det(\lambda I + j\mathcal{H}) = 0\} \\ &= -\omega_i \text{ si } j\mathcal{H} \text{ n'a aucune valeur propre positive} \end{aligned}$$

$$\begin{aligned} \delta_{\omega_i+} &= \min\{\lambda \in \mathbb{R}^+ : \det(\lambda I + j\mathcal{H}) = 0\} \\ &= \infty \text{ si } j\mathcal{H} \text{ n'a aucune valeur propre négative} \end{aligned}$$

Alors la condition (1.24) reste vraie $\forall \omega \in I(\omega_i)$ où :

$$I(\omega_i) = [\omega_i + \delta_{\omega_i-}, \omega_i + \delta_{\omega_i+}] \quad (1.27)$$

Ainsi, il suffit d'utiliser cette proposition pour plusieurs fréquences ω_i jusqu'à ce que l'union de tous les intervalles soit égale à \mathbb{R}^+ . Cette approche de type hamiltonien permet de balayer très rapidement l'ensemble des fréquences et de déterminer une borne supérieure de μ sur \mathbb{R}^+ , sans manquer de fréquences critiques. Le principe sera repris lors du chapitre 2 pour une autre caractérisation de la borne supérieure de μ .

1.2.4 Borne supérieure de μ : autres approches

La sous-section précédente a présenté la formulation la plus communément utilisée de la borne supérieure de μ . Cependant il existe de nombreuses autres méthodes, dont une grande partie sera brièvement introduite ici.

Généralisation de la formulation (D,G) par l'approche des multiplieurs

Il a été proposé dans [Fu et Barabanov 1995; 1997; Meinsma et al. 1996] une approche basée sur des multiplieurs, qui se révèle en fait être une généralisation de la formulation par matrices (D,G). Elle est basée sur le lemme suivant :

Lemme 1.1

Une famille de matrices \mathcal{A} est non-singulière s'il existe une autre matrice C , appelée multiplieur, telle que $\text{He } CA < 0$ pour tout $A \in \mathcal{A}$, avec $\text{He } X$ la partie hermitienne de la matrice X .

Ainsi, en reprenant la définition 1.2 de la valeur singulière structurée, il est clair que l'on cherche à trouver la plus grande taille k des incertitudes Δ telle que la matrice $F = I - \Delta M(j\omega_0)$ pour une fréquence ω_0 donnée est non-singulière, $\forall \Delta \in k\mathcal{B}_\Delta$, avec $k > 0$. Le lemme précédent peut donc être utilisé afin de calculer une borne supérieure de μ , par la maximisation de k tout en calculant un multiplieur non-structuré complexe C telle que $\text{He } CF < 0$. La condition précédente doit être vérifiée pour tous les Δ appartenant aux sommets de l'ensemble d'incertitudes $k\mathcal{B}_\Delta$, entraînant bien une complexité numérique exponentielle. Il est cependant prouvé que la borne qui en résulte est moins conservative que celle obtenue par la formulation (D,G), ce qui en fait une méthode intéressante pour des problèmes de taille raisonnable. De plus il existe des techniques permettant de réduire la taille du multiplieur à optimiser, ainsi que le nombre de LMIs à résoudre, afin d'améliorer le temps de calcul. Cette méthode sera largement reprise et exploitée dans le chapitre 2, afin de réduire ses défauts et également de proposer une technique pour valider la borne supérieure de μ sur le continuum fréquentiel (car c'est à la base une technique appliquée sur un maillage avec tous les inconvénients que cela entraîne).

Une extension de cette technique des multiplieurs a été proposée dans [Chen et Sugie 1996; 1997; 1998]. Plutôt que de considérer le multiplieur comme constant, il est dans cette extension considéré comme dépendant de Δ , permettant ainsi de relâcher un peu plus les contraintes et d'obtenir finalement une borne supérieure encore moins conservative. Le problème devient non-convexe, mais il est possible de se ramener à une formulation LMI grâce à [Gahinet et al. 1995]. La complexité numérique est cependant accrue, et il n'est pas proposé de façon simple de réduire le nombre de LMIs contrairement à l'approche par multiplieurs constants.

Utilisation des multiplieurs de Popov généralisés

Une autre technique basée sur les multiplieurs a également été proposée. Elle utilise les multiplieurs de Popov qui sont contenus dans une classe \mathcal{P} définis-

sant l'ensemble des matrices de transfert diagonales $P(s)$ qui satisfont $\frac{1}{2}(P(j\omega) + P^*(j\omega)) \geq 0, \forall \omega \in \mathbb{R}^+$. Ces multiplieurs ont également la propriété intéressante que pour toute matrice diagonale $\tilde{\Delta}$, $P\tilde{\Delta}$ est dans le secteur $[0, +\infty[$ si et seulement si $\tilde{\Delta}$ est dans le secteur $[0, +\infty[$. Ceci permet donc d'une certaine façon de prendre en compte la structure –réelle ou complexe– des incertitudes, pour lesquelles le multiplieur $P(s)$ a alors une structure prédéfinie. Ainsi, il est intéressant d'évoquer la transformation de secteur qui permet de passer du secteur $[-1, 1]$ sur lequel sont définies les incertitudes (ensemble \mathcal{B}_Δ), au secteur $[0, +\infty[$ sur lequel peuvent être utilisés les multiplieurs de Popov. L'avantage est un éventuel gain sur le conservatisme lorsque Δ est constitué d'incertitudes réelles. On note $\text{sectf}(X(s)) := (I - X(s)(I + X(s))^{-1})$, pour toute matrice de transfert carrée $X(s)$, cette transformation bilinéaire. Ainsi, en utilisant le lemme positif réel, le principe consiste à calculer une borne inférieure de la marge de robustesse k_{\max} directement en résolvant le problème suivant :

$$\max k \text{ tel que } \begin{cases} P(s) \in \mathcal{P} \\ \text{He}(P(j\omega) \text{sectf}(kG(j\omega))) > 0, \forall \omega \in \mathbb{R}^+ \end{cases} \quad (1.28)$$

Le multiplieur $P(s)$ peut également être pris polynomial sans restriction, ce qui amène la notion du multiplieur de Popov généralisé.

Afin de résoudre le problème d'optimisation précédent de dimension infinie, il est possible de le transformer de façon à obtenir une caractérisation LMI [Ly et al. 1994; 1998; Sparks et Bernstein 1995]. L'avantage est que cette formulation est à temps polynomial.

Remarque 1.1

Il est important de souligner que l'approche par multiplieurs de Popov est sur le fond équivalente à l'approche par matrices (D, G) . Les différences pratiques existantes entre ces deux approches concernent principalement leur validation. L'approche par matrices (D, G) est vérifiée traditionnellement fréquence par fréquence (ou par balayage fréquentiel), alors que l'approche par multiplieurs utilise le lemme Kalman Yakubovich Popov (ou KYP). Ces deux approches ne sont cependant pas exclusives de chacune des deux méthodes.

Séparateurs quadratiques

[Iwasaki et Hara 1998] propose un cadre de travail pour l'analyse de robustesse des systèmes LTI, en se basant sur la notion d'interconnexion dite bien-posée, dont la définition est donnée ci-après.

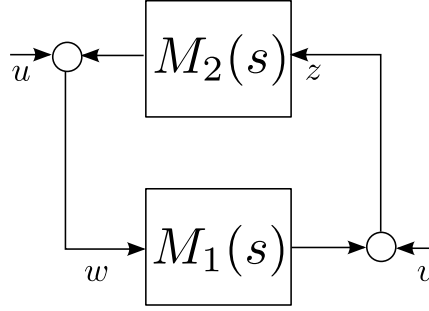


FIGURE 1.3 – Schéma d'interconnexion.

Définition 1.4

L'interconnexion $\Sigma(M_1(s), M_2(s))$ de la figure 1.3 est dite bien-posée si pour toute entrée bornée $\begin{bmatrix} v \\ u \end{bmatrix}$, il existe une unique paire $\begin{bmatrix} z \\ w \end{bmatrix}$ et un scalaire $\gamma > 0$ tels que :

$$\left\| \begin{bmatrix} z \\ w \end{bmatrix} \right\|_{\infty} \leq \gamma \left\| \begin{bmatrix} v \\ u \end{bmatrix} \right\|_{\infty} \quad (1.29)$$

Dans le cas où on se place à une fréquence donnée ω telle que $M_1 = M_1(j\omega)$, et où $M_2(s) = \Delta \in \Delta$, la notion d'interconnexion bien-posée est équivalente à la stabilité robuste de celle-ci.

Remarque 1.2

En exprimant la relation entrée-sortie de la figure 1.3, avec $M_1 = M_1(j\omega)$ et $M_2(s) = \Delta$, on a :

$$\begin{bmatrix} I & -M_1 \\ -\Delta & I \end{bmatrix} \begin{bmatrix} z \\ w \end{bmatrix} = \begin{bmatrix} v \\ u \end{bmatrix} \quad (1.30)$$

Ainsi, la condition d'existence et d'unicité de la définition d'un système bien-posé est équivalente à $\det(I - \Delta M_1) \neq 0$, montrant bien le lien avec la valeur singulière structurée (définition 1.2).

Afin de rendre la notion de système bien-posée exploitable, et également de faire le lien avec la valeur singulière structurée, le théorème suivant introduit les séparateurs quadratiques, qui donne une autre caractérisation de la stabilité robuste (en notant bien que les incertitudes sont cette fois-ci normalisées).

Théorème 1.3

Soient $M_1 = M_1(j\omega) \in \mathbb{C}^{p \times p}$ la réponse fréquentielle de $M_1(s)$ à une fréquence ω .
Les trois affirmations suivantes sont équivalentes :

1. L'interconnexion $\Sigma(M_1, \Delta)$ est bien posée pour tout $\Delta \in \mathcal{B}_\Delta$.
2. Il existe une matrice Hermitienne $\Theta \in \mathbb{C}^{2p \times 2p}$ telle que :

$$\begin{bmatrix} I & M_1 \end{bmatrix} \Theta \begin{bmatrix} I \\ M_1^* \end{bmatrix} < 0 \quad (1.31)$$

$$\begin{bmatrix} \Delta & I \end{bmatrix} \Theta \begin{bmatrix} \Delta^* \\ I \end{bmatrix} \geq 0, \quad \forall \Delta \in \mathcal{B}_\Delta \quad (1.32)$$

3. $\mu_\Delta(M_1) < 1$.

On remarque aisément que ce cadre de travail permet de faire un lien direct avec la notion des contraintes quadratiques intégrales (IQC, en anglais) [Megretski et Rantzer 1997], dans le cas où la matrice d'incertitudes est constante. Θ est ici appelé séparateur quadratique car il établit la séparation topologique du graphe de M_1 et du graphe inverse de Δ , donnant une condition nécessaire et suffisante de stabilité robuste (ou d'interconnexion bien posée). Ainsi, afin d'assurer la stabilité robuste, il est nécessaire et suffisant de trouver un séparateur quadratique $\Theta \in \Theta$ tel que $\begin{bmatrix} I & M_1 \end{bmatrix} \Theta \begin{bmatrix} I \\ M_1^* \end{bmatrix} < 0$, où :

$$\Theta := \left\{ \Theta : \Theta = \Theta^*, \begin{bmatrix} \Delta & I \end{bmatrix} \Theta \begin{bmatrix} \Delta^* \\ I \end{bmatrix} \geq 0, \forall \Delta \in \mathcal{B}_\Delta \right\} \quad (1.33)$$

et la valeur singulière structurée est ainsi donnée par :

$$\mu_\Delta(M_1) = \inf_{\gamma > 0, \Theta \in \Theta} \left\{ \gamma : \begin{bmatrix} \gamma I & M_1 \end{bmatrix} \Theta \begin{bmatrix} \gamma I \\ M_1^* \end{bmatrix} < 0 \right\} \quad (1.34)$$

Bien qu'étant un problème de faisabilité convexe, il est difficile à résoudre à cause de l'ensemble Θ . En prenant un ensemble plus exploitable compris dans Θ , il est possible de rendre le problème faisable, mais ceci introduit également du conservatisme et donne donc une borne supérieure de $\mu_\Delta(M_1)$. La sous-classe $\Theta_v \subset \Theta$ proposée par [Iwasaki et Hara 1998], dans le cas d'incertitudes purement réelles

$\Delta \in \mathcal{B}_\Delta := \{\text{diag}(\delta_1 I_{k_1}, \dots, \delta_r I_{k_r}), \delta_i \in \mathbb{R}, |\delta_i| \leq 1\}$, est la suivante :

$$\Theta_v(\mathcal{B}_\Delta) := \left\{ \Theta = \begin{bmatrix} R & S \\ S^T & Q \end{bmatrix} : R_{ii} \leq 0, i = 0, \dots, r, \right. \\ \left. \begin{bmatrix} \Delta_j & I \end{bmatrix} \Phi \begin{bmatrix} \Delta_j \\ I \end{bmatrix} \geq 0, j = 1, \dots, 2^r \right\} \quad (1.35)$$

où R , S et Q sont des matrices réelles avec $R = R^T$, $Q = Q^T$, R_{ii} représente le i -ème bloc de taille $k_i \times k_i$ situé sur la diagonale de R , et Δ_j ($j = 1, \dots, 2^r$) correspondent aux sommets de \mathcal{B}_Δ .

Remarque 1.3

Une fois encore, il existe un fort lien entre cette approche et l'approche par matrices (D, G) . En effet, en choisissant la sous-classe :

$$\Theta_{DG} := \left\{ \Theta = \begin{bmatrix} -D & G \\ G^* & D \end{bmatrix} : D \in \mathcal{D}_n, G \in \mathcal{G}_n \right\} \quad (1.36)$$

on retombe sur la caractérisation présentée à la section 1.2.3. On prouve également que la borne supérieure de μ obtenue en utilisant la sous-classe Θ_v est au moins aussi bonne que celle utilisant Θ_{DG} , montrant l'avantage en terme de conservatisme de l'approche par séparateurs quadratiques telle que présentée ici.

Même si cette méthode permet à une fréquence donnée d'obtenir un conservatisme moindre que la méthode classique par matrices (D, G) , il faut cependant être conscient qu'il s'agit d'une méthode de résolution à temps exponentiel, puisque le nombre de contraintes croît exponentiellement avec le nombre d'incertitudes. De plus si l'on souhaite se ramener à un problème de dimension finie pour évaluer une borne valable sur toute la plage de fréquence, le lemme de Kalman-Yakubovich-Popov (KYP, généralisé dans [Anderson 1967]) introduit de façon équivalente un problème LMI de taille importante.

D-stabilité et *Stability Feeler*

La D-stabilité implique que tous les zéros d'une famille de polynômes caractéristiques soient dans une région ouverte imposée \mathcal{D} , symétrique par rapport à l'axe des réels du plan complexe (par exemple, le demi-plan gauche du plan complexe). En réécrivant $\det(I - M(s)\Delta)$ comme une série de polynômes réels, [Matsuda et al. 2009; Matsuda et Mori 2009] proposent d'étudier la stabilité de systèmes incertains LTI avec l'introduction de leur outil nommé *Stability Feeler*. Le principe ne sera pas rappelé ici puisque la méthode a brièvement été expliquée à la sous-section concernant le calcul d'une borne inférieure de μ (voir sous-section 1.2.2, puis pa-

ragraphe "Méthodes de complexité exponentielle"). Il convient juste d'insister sur le fait qu'il s'agit d'une méthode à temps exponentiel, qui peut également utiliser des techniques d'expansion ou de division type branch-&-bound de l'espace d'incertitudes, afin de réduire le conservatisme. Cette méthode peut donc s'avérer intéressante pour la précision de la borne supérieure obtenue, mais elle est de complexité exponentielle et donc d'usage limité.

Évaluation sans le formalisme LFT

Récemment, il a été proposé par [Zhao et al. 2011] une approche pour évaluer les bornes inférieure et supérieure de μ sans avoir à mettre le système incertain sous la forme LFR (Figure 1.2). Ceci peut être intéressant lorsqu'il n'est pas possible d'isoler les incertitudes car elles n'apparaissent pas de façon rationnelle. On peut par exemple citer le cas d'une dépendance trigonométrique des paramètres incertains, où seules des approximations polynomiales permettent d'obtenir une LFR, au prix d'une modification du problème initial.

Pour un système incertain dont la dynamique est exprimée par $\dot{x} = A(\delta)x$, avec δ le vecteur d'incertitudes, le principe consiste à écrire $\dot{x} = A(0)x + A_\Delta(\delta)x$, avec $A_\Delta(\delta) := A(\delta) - A(0)$, puis à évaluer $A_\Delta(\delta)$ pour un certain nombre de valeurs de δ appartenant à l'espace d'incertitudes admissibles. L'analyse de robustesse est ensuite exprimée par la recherche de la plus petite perturbation δ telle que $|I - M(j\omega)A_\Delta(\delta)| = 0$, pour tout $\omega \in \mathbb{R}^+$ et avec $M(j\omega) = (j\omega I - A(0))^{-1}$. Une approche géométrique similaire à celle de [Kim et al. 2008] pour la borne inférieure est utilisée, avec un tirage aléatoire pour évaluer plusieurs $A_\Delta(\delta)$, ainsi que des étapes de contraction et d'expansion pour approcher la région singulière de l'espace d'incertitudes.

Cette méthode est donc fortement dépendante du nombre de tirages réalisés, tant pour la précision que pour le temps de calcul, ce dernier pouvant vite devenir prohibitif. Elle doit de plus être réalisée sur un maillage fréquentiel. Elle peut cependant être intéressante dans le cas où il est difficile d'obtenir une LFR. Il faut également noter que la méthode est applicable dans le cas d'incertitudes variant dans le temps.

1.3 DU CONSERVATISME DE LA BORNE SUPÉRIEURE DE μ

Dans l'étude proposée par [Fabrizi et al. 2014] qui a consisté à tester de nombreuses méthodes de calcul de borne inférieure de μ sur un nombre conséquent de benchmarks réalistes, il a été montré que les algorithmes existants sont généralement très satisfaisants. On en tire notamment les conclusions suivantes :

- Dans le cas purement complexe ou mixte, le *power algorithm* de [Young et Doyle 1997] donne d'excellents résultats, fournissant quasiment à chaque fois la meilleure borne inférieure en un temps de calcul très faible. Des améliorations ont de plus été introduites dans la boîte à outils Matlab SMAC ([SMAC 2012-2015]). Elles consistent à appliquer une première fois le *power algorithm* sur un maillage fréquentiel grossier, puis à raffiner le maillage autour des fréquences pour lesquelles la borne inférieure est la plus élevée.
- Dans le cas purement réel, la technique de migration des pôles de [Ferrerres et Biannic 2001] est de loin la plus efficace, que ce soit en précision ou en temps de calcul. Des améliorations ont également été proposées pour augmenter encore plus la précision, en combinant plusieurs méthodes selon les trois étapes suivantes :
 1. La technique de migration de pôles est tout d'abord appliquée,
 2. Le *gain-based algorithm* de [Seiler et al. 2010] est ensuite exécuté pour quelques fréquences en utilisant les résultats de l'étape précédente comme initialisation. Ces fréquences peuvent être celles pour lesquelles le système est instable suite à l'application de la technique précédente, ou bien celles pour lesquelles la plus grande borne supérieure de μ est obtenue.
 3. Des outils d'optimisation par essaim de particules sont finalement utilisés, en utilisant les résultats précédents comme initialisation.

Cette stratégie est également implantée dans la boîte à outils Matlab SMAC ([SMAC 2012-2015]).

Il existe donc des algorithmes efficaces pour calculer une borne inférieure de μ dans quasiment tous les cas de figures en un temps très raisonnable. Quant à la borne supérieure, la caractérisation présentée à la sous-section 1.2.3 est la plus communément utilisée, avec des routines présentes notamment dans la Robust Control Toolbox de Matlab [The MathWorks 2010]. Dans de nombreux cas, elle s'avère être pertinente pour fournir une borne supérieure de μ de façon rapide. Les autres méthodes présentées à la sous-section 1.2.4 permettent certes d'améliorer la précision de la borne obtenue, mais ne sont viables que pour des systèmes assez petits puisqu'elles possèdent bien souvent une complexité exponentielle en fonction du nombre de paramètres incertains. Cependant, un certain nombre de benchmarks issus de la littérature et regroupés dans l'étude de [Fabrizi et al. 2014], révèlent l'existence d'un écart plus ou moins important entre les bornes inférieure et supérieure lorsque la formulation (D,G) est utilisée pour la borne supérieure. Il est important de souligner que ces benchmarks ont la particularité d'être constitué uniquement d'incertitudes paramétriques réelles, puisque dans le cas général le conservatisme

est faible en présence d'incertitudes complexes. La liste des benchmarks considérés est présentée dans le tableau 1.1, avec :

- la description du système,
- le nombre d'états,
- la structure de la matrice d'incertitudes, selon la forme $m \times p$, signifiant qu'il y a m blocs réels diagonaux de taille $p \times p$.
- la meilleure borne inférieure $\underline{\mu}_\Delta$ parmi tous les algorithmes existants,
- la meilleure borne supérieure $\bar{\mu}_\Delta$ par la formulation des matrices (D,G),
- l'écart η entre les bornes inférieure et supérieure.

N°	Description	États	Structure de Δ	$\underline{\mu}_\Delta$	$\bar{\mu}_\Delta$	Écart η
1	Exemple académique	4	2×2	1.2517	2.4131	92.79%
2	Véhicule aéronautique	8	4×1	1.6200	1.9757	21.96%
3	Véhicule "Drive-by-wire"	4	2×1+7×2	0.9966	1.1332	13.71%
4	Véhicule de rentrée	7	3×1+1×4+1×6	0.9947	1.0156	2.10%
5	Avion rigide	9	14×1	0.1842	0.2283	23.94
6	Avion militaire	10	7×1+1×2+1×3+1×15	1.6668	2.1572	29.42%
7	Télescope	70	20×1	46.3722	46.5977	0.49%
8	Disque dur	29	19×1+4×2	1.2134	1.2144	0.00%
9	Lanceur	30	16×1+10×2+1×3+1×6	0.8619	2.7232	215.95%
10	Réseau biochimique	7	13×39	733.4645	899.98	22.70%
11	Satellite	95	3×1+8×2+1×3+3×4	0.1006	0.3839	281.61%

TABLEAU 1.1 – Liste des benchmarks considérés.

On remarque que sur l'ensemble de ces benchmarks, l'écart est effectivement plus ou moins important.

Remarque 1.4

Pour les benchmarks N° 4, 7 et 8, l'écart est relativement faible. Cependant le problème réside dans le temps de calcul de la borne supérieure qui a nécessité d'utiliser un solveur LMI. En effet, l'utilisation d'un algorithme de descente de gradient, bien que plus rapide, donne une borne supérieure très sous-optimale.

Remarque 1.5

Pour tous les benchmarks du tableau 1.1, les incertitudes sont normalisées. Ainsi, leur stabilité robuste est prouvée si $\mu_\Delta < 1$.

Une question légitime peut maintenant se poser : quelle borne est généralement responsable d'un écart conséquent entre les bornes supérieure et inférieure de μ ? Afin d'introduire et de justifier les travaux qui vont suivre, un algorithme de branch-&-bound, introduit par [Newlin et Young 1997] dans le cadre de la μ -

analyse et permettant théoriquement de faire tendre l'écart entre les bornes vers zéro, révèle que la borne supérieure est quasiment tout le temps sur-évaluée. La borne inférieure reste en effet inchangée, alors que la borne supérieure diminue au fur et à mesure que le domaine d'incertitudes est partitionné. Bien entendu, l'utilisation systématique du branch-&-bound n'est pas conseillée dans sa version classique, puisque le temps de calcul augmente exponentiellement en fonction du nombre d'incertitudes. Ce problème sera traité lors du chapitre 2.

CONCLUSION DU CHAPITRE

Dans ce chapitre, les notions clés concernant la μ -analyse ont été introduites. Cette technique d'analyse de robustesse des systèmes LTI est aujourd'hui bien mûre et permet de traiter de nombreux problèmes. Au fil des années de nouveaux algorithmes sont apparus permettant d'évaluer avec plus de précision les bornes inférieure et supérieure de la valeur singulière structurée. Cependant il existe toujours des cas plus ou moins pathologiques, où soit l'écart entre les bornes est trop important, soit le temps de calcul nécessaire pour évaluer la borne supérieure est excessif.

De façon empirique, il a été montré que les algorithmes servant à évaluer une borne inférieure donnent bien souvent la valeur exacte de μ . Le conservatisme semble donc dû au calcul de la borne supérieure lorsque la formulation classique par matrices (D,G) est utilisée. Les autres algorithmes qui ont été présentés brièvement dans ce chapitre ont soit un temps de calcul à complexité exponentielle, soit n'apportent aucun avantage en terme de conservatisme par rapport à la formulation (D,G) . La méthode des multiplieurs de [Fu et Barabanov 1997] sera cependant exploitée dans la suite, car elle permet une maîtrise du temps de calcul et est toujours moins conservative que l'approche classique.

Dans ce contexte, le prochain chapitre présentera une première partie des contributions de ces travaux de thèse. Il proposera trois nouvelles méthodes permettant soit de réduire le temps de calcul nécessaire pour calculer une borne supérieure de μ lorsque le conservatisme est faible, soit de diminuer l'écart avec la borne inférieure. Les algorithmes utilisés serviront ensuite lors de la deuxième partie de ce manuscrit traitant de la synthèse de lois de pilotage.

TECHNIQUES DE RÉDUCTION DU CONSERVATISME DE LA BORNE SUPÉRIEURE DE μ

SOMMAIRE

2.1	INTRODUCTION AUX μ -SENSIBILITÉS	42
2.1.1	Définition des μ -sensibilités	42
2.1.2	Considérations numériques	43
2.2	AMÉLIORATION DES APPROCHES BASÉES SUR LA FORMULATION PAR MA- TRICES (D,G)	48
2.2.1	Résolution partielle des LMIs de la caractérisation par matrices (D,G)	48
2.2.2	Nouvel algorithme de branch-&-bound	51
2.3	UNE BORNE SUPÉRIEURE DE μ MOINS CONSERVATIVE PAR L'APPROCHE PAR MULTIPLIEURS	53
2.3.1	Rappels de l'approche	54
2.3.2	μ -sensibilités et réduction du temps de calcul	57
2.3.3	Validation des multiplieurs sur de larges intervalles fréquentiels . .	59
2.3.4	Résultats	63
	CONCLUSION	66

Résumé

TROIS nouvelles méthodes sont proposées dans ce chapitre afin de réduire l'écart entre la valeur singulière structurée μ et ses bornes supérieures, tout en maintenant un temps de calcul raisonnable. La première est basée sur la caractérisation classique par matrices (D,G), la deuxième sur une approche plus générale à base de multiplieurs, et la troisième sur une technique de type branch-&-bound. Toutes ces méthodes utilisent les μ -sensibilités pour sélectionner les incertitudes ayant le plus d'influence sur la marge de stabilité, permettant soit d'obtenir des LMI avec un nombre limité de variables de décision, soit de subdiviser intelligemment le domaine d'incertitudes. Elles sont testées avec succès sur une liste de benchmarks réalistes.

La μ -analyse, introduite lors du précédent chapitre, est considérée aujourd'hui comme une technique efficace pour évaluer la stabilité robuste d'une très large classe de systèmes LTI, en présence d'incertitudes paramétriques et de dynamiques négligées. Cependant, étant donné que le calcul de la valeur singulière structurée est un problème NP-difficile, seules des bornes de cette dernière sont obtenues. La borne supérieure fournit une marge de stabilité garantie, souvent pessimiste, du système. La borne inférieure quant à elle permet d'évaluer le conservatisme de la borne supérieure.

L'étude de [Fabrizi et al. 2014], reprise partiellement au précédent chapitre, a montré que les méthodes existantes pour le calcul d'une borne inférieure se révèlent généralement très satisfaisantes (voir aussi [Ferrerres et Biannic 2001]). Les résultats sont plus contrastés pour ce qui est de la borne supérieure. La caractérisation classique par matrices (D,G) peut être évaluée de façon rapide en utilisant un algorithme de descente de gradient, mais elle s'avère parfois très conservative. Des méthodes existent pour améliorer ce point, comme par exemple :

- utiliser un solveur LMI pour résoudre le problème d'optimisation lié à la caractérisation par matrices (D,G) de [Young et al. 1995],
- employer l'approche moins conservative dite des multiplieurs de [Fu et Barabanov 1997],
- utiliser une technique de branch-&-bound comme proposé dans [Newlin et Young 1997].

Cependant ces méthodes, bien que pouvant fortement réduire le conservatisme de la borne supérieure, présentent un problème commun : le temps de calcul peut devenir prohibitif lorsque le nombre d'incertitudes est trop élevé. Afin de pallier cette difficulté, une idée consiste à se concentrer sur un nombre réduit d'incertitudes. Ainsi, en optimisant les matrices (D,G) ou les multiplieurs uniquement pour certaines incertitudes bien choisies, ou en subdivisant le domaine d'incertitudes suivant certaines directions seulement pour le branch-&-bound, on peut s'attendre à une amélioration sensible du temps de calcul sans trop dégrader le conservatisme. Les μ -sensibilités, introduites par [Braatz et Morari 1991], permettent de déterminer ces incertitudes, et sont la clé de voûte de ce chapitre dédié à l'amélioration des méthodes évoquées ci-dessus.

Le chapitre est organisé de la façon suivante. La section 2.1 présente les μ -sensibilités et la façon de les calculer. La section 2.2 introduit deux nouveaux algorithmes pour diminuer le conservatisme de la borne supérieure de μ_Δ lorsque l'approche par les matrices (D,G) est utilisée. La section 2.3 détaille l'approche moins conservative des multiplieurs, en proposant un algorithme qui veille à la fois à réduire le temps de calcul et à valider la borne obtenue sur l'ensemble de la plage de fréquences sans utiliser de maillage fréquentiel. Enfin, la dernière section

conclut en insistant tout particulièrement sur l'intérêt d'utiliser les μ -sensibilités pour toutes ces méthodes.

Remarque 2.1

Dans tout ce chapitre, et afin d'éviter d'éventuelles confusions, $\bar{\mu}_\Delta(M)$ dénote la borne supérieure de μ évaluée à une fréquence donnée ω_0 , à l'aide de l'approche par matrices (D, G) , avec $M = M(j\omega_0)$, alors que $\bar{\mu}_\Delta$ correspond à $\sup_{\omega \in \mathbb{R}^+} \mu_\Delta(M(j\omega))$, i.e. à l'inverse de la marge de robustesse k_{max} (voir définition 1.3).

2.1 INTRODUCTION AUX μ -SENSIBILITÉS

On s'intéresse dans cette section à la notion de μ -sensibilités, qui servira dans chacun des algorithmes proposés par la suite pour améliorer le conservatisme de la borne supérieure de μ_Δ ainsi que son temps de calcul.

2.1.1 Définition des μ -sensibilités

Les μ -sensibilités ont été introduites par [Braatz et Morari 1991], afin d'identifier quelles parties d'un système doivent être mieux modélisées pour améliorer son identification, en vue d'une synthèse de loi de commande robuste. Si cette dernière ne permet pas d'obtenir une boucle fermée stable (on a alors $k_{max} < 1$ dans le cas d'incertitudes normalisées), soit les spécifications en performance ne sont pas adaptées, soit le modèle n'est pas suffisamment affiné pour la synthèse. Dans ce dernier cas, comment être sûr que les incertitudes mieux identifiées apporteront une amélioration significative pour la synthèse de la loi? C'est dans ce contexte que les μ -sensibilités ont été introduites, et leur définition est rappelée ci-après.

Remarque 2.2

Afin de ne pas surcharger les notations, on se place dans le cas où les incertitudes sont uniquement paramétriques réelles, potentiellement répétées, de telle sorte que le bloc d'incertitudes Δ soit de la forme :

$$\Delta = \text{diag}(\delta_1 I_{k_1}, \dots, \delta_r I_{k_r}) \quad (2.1)$$

où les δ_i sont des scalaires réels k_i -répétés. Il faut insister sur le fait que c'est de toute façon la configuration la plus critique en termes de conservatisme de la borne supérieure de μ_Δ . On observe en effet en pratique que la présence de blocs complexes tend à le diminuer.

Définition 2.1

Soit $M = M(j\omega)$ la réponse fréquentielle de $M(s)$ à une certaine fréquence ω . On suppose que Δ est pré- et post-multiplié par une matrice diagonale $A(\epsilon_j) = \text{diag}(I_{k_1}, \dots, (1 - \epsilon_j)I_{k_j}, \dots, I_{k_r})$, de façon que la $j^{\text{ème}}$ incertitude est réduite d'un petit facteur $(1 - \epsilon_j)^2$ alors que les autres gardent leur valeur nominale. La $j^{\text{ème}}$ μ -sensibilité est alors définie par :

$$S_j^\mu = \frac{\partial}{\partial \epsilon_j} \bar{\mu}_\Delta(M_{\epsilon_j}) = \lim_{\epsilon_j \rightarrow 0^+} \frac{|\bar{\mu}_\Delta(M) - \bar{\mu}_\Delta(M_{\epsilon_j})|}{\epsilon_j} \quad (2.2)$$

avec $M_{\epsilon_j} = A(\epsilon_j)MA(\epsilon_j)$ (en remarquant que $M_0 = M$), et $\bar{\mu}_\Delta(\cdot)$ la borne supérieure évaluée à une fréquence à l'aide de l'approche par matrices (D, G) .

Remarque 2.3

Puisque le calcul exact de la valeur singulière structurée $\mu_\Delta(M_{\epsilon_j})$ (ou $\mu_\Delta(M)$) est un problème NP-difficile, et que dans ces travaux de thèse seul le calcul de la borne supérieure nous intéresse, la définition 2.1 est adaptée ici, présentant les μ -sensibilités comme la dérivée de la borne supérieure $\bar{\mu}_\Delta(M_{\epsilon_j})$ par rapport à ϵ_j .

La signification concrète des μ -sensibilités est la suivante : plus la valeur de S_j^μ est élevée, plus l'incertitude correspondante a de l'influence sur la valeur de $\bar{\mu}_\Delta(M)$. Dans l'étude de [Braatz et Morari 1991], les plus fortes μ -sensibilités indiquent les incertitudes les plus critiques pour le système modélisé, sur lesquelles il faudra améliorer les connaissances afin d'augmenter la marge de stabilité et le niveau de performance lors de la synthèse.

Dans le contexte de cette thèse, les incertitudes possédant les μ -sensibilités les plus élevées sont a priori les plus responsables du conservatisme de $\bar{\mu}_\Delta(M)$. Les contributions utilisant cet outil afin d'améliorer l'écart entre μ_Δ et ses bornes supérieures en un temps raisonnable seront détaillées dans les sections 2.2 et 2.3. Cependant, il reste tout d'abord à expliciter un peu mieux le calcul des μ -sensibilités, la formulation (2.2) étant sensible numériquement. La sous-section suivante vise à donner quelques informations à ce sujet.

2.1.2 Considérations numériques

L'utilisation de la formulation (2.2) repose sur le calcul de différences finies, en choisissant un ϵ_j le plus faible possible. Le problème inhérent à cette méthode est sa forte sensibilité numérique. En effet, elle amène à faire le rapport entre une différence de nombres proches mais assez grands, par un nombre très faible. Étant donné de plus que le calcul de $\bar{\mu}_\Delta(M)$ peut s'avérer inexact, les valeurs des μ -sensibilités peuvent être faussées, et il peut être difficile de choisir une valeur ap-

propriété pour ϵ_j . Fort heureusement, une solution a été apportée dans [Douglas et Athans 1995] qui relève d'ailleurs ce problème numérique, et cette sous-section vise à détailler les grandes lignes de l'algorithme proposé.

La première étape consiste à choisir une fréquence à laquelle le calcul des μ -sensibilités sera réalisé. La fréquence la plus naturelle est celle où la valeur de $\bar{\mu}_\Delta$ est maximale. On note alors ω_c cette fréquence, et $M = M(j\omega_c)$ la réponse fréquentielle associée. On introduit également $M_{\epsilon_j} = A(\epsilon_j)MA(\epsilon_j)$, selon les notations de la définition 2.1. Le but est donc de calculer la dérivée partielle suivante :

$$\frac{\partial}{\partial \epsilon_j} \bar{\mu}_\Delta(M_{\epsilon_j}) \quad (2.3)$$

sans utiliser l'approximation des différences finies. Pour ce faire, on réécrit la formulation (1.21) du chapitre 1 de façon à se ramener à un problème de calcul de plus grande valeur propre :

$$\bar{\mu}_\Delta(M_{\epsilon_j}) = \inf_{D \in \mathcal{D}_n, G \in \mathcal{G}_n} \left[\min_{\beta \geq 0} \{ \beta : \bar{\lambda}(M_{D_{\epsilon_j}}^* M_{D_{\epsilon_j}} + j(GM_{D_{\epsilon_j}} - M_{D_{\epsilon_j}}^* G)) \leq \beta^2 \} \right] \quad (2.4)$$

où $\bar{\lambda}(\cdot)$ représente la plus grande valeur propre, et $M_{D_{\epsilon_j}} = DA(\epsilon_j)MA(\epsilon_j)D^{-1}$.

On suppose qu'il existe, pour chaque ϵ_j , des matrices \bar{D} et \bar{G} telles que l'infimum est atteint, et on note :

$$(\bar{\beta}(\epsilon_j))^2 = \bar{\lambda}(M_{\bar{D}_{\epsilon_j}}^* M_{\bar{D}_{\epsilon_j}} + j(\bar{G}M_{\bar{D}_{\epsilon_j}} - M_{\bar{D}_{\epsilon_j}}^* \bar{G})) \quad (2.5)$$

Afin d'introduire globalement la méthode proposée par [Douglas et Athans 1995] pour calculer les μ -sensibilités sans surcharger les notations, on considère le problème précédent comme un problème du type :

$$\min_x f(x, \epsilon_j) \quad (2.6)$$

En notant $\bar{x}(\epsilon_j)$ la valeur qui permet d'atteindre le minimum global de la fonction pour un ϵ_j donné, le but est donc de calculer :

$$\frac{d}{d\epsilon_j} f(\bar{x}(\epsilon_j), \epsilon_j) \quad (2.7)$$

Le problème est qu'il n'est pas possible de connaître la variation de la fonction $\epsilon_j \mapsto \bar{x}(\epsilon_j)$, et donc une telle dérivée est ici impossible à évaluer simplement. Il existe cependant un lemme qui permet de contourner le problème, et dont les hypothèses ne sont aucunement restrictives (voir [Douglas et Athans 1995]) :

Lemme 2.1

On suppose que :

$$\nabla_x f(x, \epsilon_j) \Big|_{x=\bar{x}(\epsilon_j)} = 0 \quad (2.8)$$

$$\nabla_x^2 f(x, \epsilon_j) \Big|_{x=\bar{x}(\epsilon_j)} > 0 \quad (2.9)$$

où ∇_x est l'opérateur gradient selon x , et ∇_x^2 le hessien. Alors :

$$\frac{d}{d\epsilon_j} f(\bar{x}(\epsilon_j), \epsilon_j) = \frac{\partial}{\partial \epsilon_j} f(x, \epsilon_j) \Big|_{x=\bar{x}(\epsilon_j)} \quad (2.10)$$

En prenant x comme un vecteur contenant les variables d'optimisation des matrices D et G , et $f(x, \epsilon_j) = \bar{\mu}(M_{\epsilon_j})$, le lemme précédent est alors applicable. Il reste ainsi à déterminer les dérivées partielles de valeurs propres d'un opérateur symétrique linéaire. Il faut souligner tout d'abord que de telles dérivées existent, comme il est prouvé dans [Douglas et Athans 1995] ou dans le chapitre 2 de [Kato 1976]. Leur calcul peut ensuite être réalisé grâce au théorème suivant dont la preuve consiste en de simple manipulations algébriques et suit celle donnée dans [Freudenberget al. 1982].

Théorème 2.1 ([Douglas et Athans 1995])

Soient $\alpha \in \mathbb{R}$ appartenant à un voisinage $\mathcal{B}(\bar{\alpha})$ ouvert de $\bar{\alpha} \in \mathbb{R}$, et $X(\alpha) \in \mathbb{C}^{n \times n}$ une matrice symétrique. Soit λ_1 une valeur propre de $X(\bar{\alpha})$ répétée m fois. Pour tout α appartenant au voisinage $\mathcal{B}(\bar{\alpha})$, $X(\alpha)$ admet une décomposition en éléments propres associée $X(\alpha) = U(\alpha)\Lambda(\alpha)U^*(\alpha)$, telle que :

$$U(\alpha) = \begin{bmatrix} U_1(\alpha) & U_2(\alpha) \end{bmatrix}, \quad U_1(\alpha) \in \mathbb{C}^{n \times m}, \quad U_2(\alpha) \in \mathbb{C}^{n \times (n-m)} \quad (2.11)$$

$$\Lambda = \begin{bmatrix} \Lambda_1(\alpha) & 0 \\ 0 & \Lambda_2(\alpha) \end{bmatrix}, \quad \Lambda_1(\alpha) \in \mathbb{C}^{m \times m}, \quad \Lambda_2(\alpha) \in \mathbb{C}^{(n-m) \times (n-m)} \quad (2.12)$$

où $\Lambda_1(\bar{\alpha}) = \lambda_1 I_m$. Alors les dérivées des valeurs propres de $\Lambda_1(\alpha)$ évaluées en $\alpha = \bar{\alpha}$ sont données par les valeurs propres de la matrice :

$$U_1^*(\bar{\alpha}) \frac{\partial}{\partial \alpha} X(\alpha) \Big|_{\alpha=\bar{\alpha}} U_1(\bar{\alpha}) \quad (2.13)$$

Ainsi, dans notre contexte, en remplaçant α par ϵ_j contenu dans un voisinage de 0, et la matrice $X(\alpha)$ par :

$$M_{D_{\epsilon_j}}^* M_{\bar{D}_{\epsilon_j}} + j(\bar{G} M_{\bar{D}_{\epsilon_j}} - M_{D_{\epsilon_j}}^* \bar{G}) \quad (2.14)$$

on obtient un moyen de calculer les dérivées selon ϵ_j des plus grandes valeurs propres de la matrice définie par (2.14), autrement dit les dérivées de $\bar{\beta}^2(\epsilon_j)$ (voir équation (2.5)). L'algorithme suivant est alors proposé afin de calculer les μ -sensibilités sans utiliser les différences finies.

Algorithme 1: Calcul des μ -sensibilités

Entrées : M, Δ

Sorties : $S^\mu = \begin{bmatrix} S_1^\mu & \dots & S_r^\mu \end{bmatrix}^T$

- 1 Résoudre (2.4) pour $\epsilon_j = 0$. Soient \bar{D} et \bar{G} les matrices optimales obtenues, et $\bar{\beta}$ la valeur de β correspondante ;
- 2 Calculer la structure propre composée de U et Λ telle que :

$$U\Lambda U^* = M_{\bar{D}}^* M_{\bar{D}} + j(\bar{G} M_{\bar{D}} - M_{\bar{D}}^* \bar{G}) \quad (2.15)$$

où $M_{\bar{D}} = \bar{D} M \bar{D}^{-1}$. On suppose que la plus grande valeur propre est répétée m fois, et on note U_1 la matrice constituée des m premières colonnes de U ;

pour $j = 1$ à r

- 3 Définir la matrice $A(\epsilon_j)$ comme dans la définition 2.1 ;
- 4 Définir la matrice Z , correspondant à la dérivée selon ϵ_j de la matrice exprimée par (2.14), évaluée en $\epsilon_j = 0$:

$$Z = \frac{\partial}{\partial \epsilon_j} M_{\bar{D}_{\epsilon_j}}^* \Big|_{\epsilon_j=0} M_{\bar{D}} + M_{\bar{D}}^* \frac{\partial}{\partial \epsilon_j} M_{\bar{D}_{\epsilon_j}} \Big|_{\epsilon_j=0} + j \left(\bar{G} \frac{\partial}{\partial \epsilon_j} M_{\bar{D}_{\epsilon_j}} \Big|_{\epsilon_j=0} - \frac{\partial}{\partial \epsilon_j} M_{\bar{D}_{\epsilon_j}}^* \Big|_{\epsilon_j=0} \bar{G} \right)$$

- 5 En utilisant le résultat du théorème 2.1, calculer :

$$S_j^\mu = \frac{\bar{\lambda}(-U_1^* Z U_1)}{2\bar{\beta}} \quad (2.16)$$

où $\bar{\lambda}$ représente la plus grande valeur propre ;

- 6 $S^\mu = \begin{bmatrix} S_1^\mu & \dots & S_r^\mu \end{bmatrix}^T$ est un vecteur constitué des μ -sensibilités de chacune des r incertitudes ;
-

Remarque 2.4

Quelques explications méritent d'être fournies notamment concernant l'étape 5 de l'algorithme 1, qui donne l'expression de la $j^{\text{ème}}$ μ -sensibilité :

- le signe négatif représente le fait que l'on s'intéresse à la décroissance de $\bar{\mu}_\Delta(M_{\epsilon_j})$, afin de respecter la convention de la définition 2.1 où chaque incertitude est réduite d'un petit facteur,

- la plus grande valeur propre est considérée puisque toutes les valeurs propres ne décroissent pas à la même vitesse, et on s'intéresse ici à celle qui décroît le plus,
- enfin la division par $2\bar{\beta}$ provient du fait que l'on souhaite obtenir la μ -sensibilité de $\bar{\mu}_\Delta(M)$ et non de $\bar{\mu}_\Delta^2(M)$.

L'algorithme 1 est ensuite comparé avec les différences finies sur le benchmark 5 du tableau 1.1, correspondant à un modèle d'avion rigide possédant 14 incertitudes non-répétées. Les résultats sont donnés dans le tableau 2.1. Les valeurs des μ -sensibilités sont relativement proches entre les méthodes, mais il est important de noter que l'utilisation des différences finies peut amener des écarts importants lorsque la valeur de ϵ_j est changée, l'incertitude possédant la plus grande μ -sensibilité pouvant même être différente d'un cas à l'autre. Ici on observe que l'incertitude δ_{14} est la plus influente sur la valeur de $\bar{\mu}_\Delta(M)$, mais un ϵ_j trop faible tend à rendre l'incertitude δ_9 plus influente, confirmant bien la grande sensibilité numérique des différences finies vis-à-vis du choix de ϵ_j . Enfin, toujours pour les différences finies et pour avoir de bons résultats, il faut une estimation très précise de $\bar{\mu}_\Delta(M)$, entraînant un temps de calcul souvent bien plus important que la méthode proposée par l'algorithme 1.

Incertain-tude	Calcul des μ -sensibilités		
	Différences finies		Algorithme 1
	$\epsilon_j = 10^{-4}$	$\epsilon_j = 10^{-6}$	
δ_1	0.0372	0.0300	0.0272
δ_2	0.0087	0.0080	0.0053
δ_3	0.0264	0.0281	0.0044
δ_4	0.0132	0.0087	0.0070
δ_5	0.0369	0.0322	0.0258
δ_6	0.0388	0.0407	0.0289
δ_7	0.0422	0.0422	0.0537
δ_8	0.0255	0.0254	0.0197
δ_9	0.0756	0.0839	0.0735
δ_{10}	0.0185	0.0161	0.0168
δ_{11}	0.0062	0.0063	0.0021
δ_{12}	0.0429	0.0432	0.0373
δ_{13}	0.0111	0.0111	0.0048
δ_{14}	0.0913	0.0817	0.0852

TABLEAU 2.1 – Calcul des μ -sensibilités pour le benchmark N° 5 de l'avion rigide : comparaison entre l'utilisation des différences finies et de l'algorithme 1

On se propose lors des paragraphes suivants d'utiliser ces notions de μ -sensibilités pour améliorer le calcul d'une borne supérieure de μ_Δ .

2.2 AMÉLIORATION DES APPROCHES BASÉES SUR LA FORMULATION PAR MATRICES (D,G)

Il est proposé dans cette section d'améliorer le calcul de la borne supérieure de μ_Δ utilisant la formulation par matrices (D,G), introduite à la proposition 1.1 du chapitre précédent. Cette caractérisation communément utilisée est attractive car elle permet, en utilisant une technique de descente de gradient, d'obtenir une borne supérieure très rapidement. Cependant, cette dernière s'avère dans certains cas très conservative. Il est alors possible :

- d'utiliser un solveur LMI et la formulation (1.21),
- d'employer une technique de branch-&-bound comme proposé dans [Newlin et Young 1997].

Cependant ces deux méthodes augmentent sensiblement le temps de calcul qui peut vite s'avérer prohibitif dans le cas d'un trop grand nombre d'incertitudes. Cette section propose d'améliorer ce point sur chacune de ces deux méthodes, en utilisant notamment les μ -sensibilités introduites lors du paragraphe précédent.

Remarque 2.5

Les calculs présentés dans les pages suivantes ont été réalisés sur un ordinateur possédant 6 Go de RAM, et un processeur quatre cœurs Intel® Xeon(R) W3530 cadencé à 2.80Ghz, tournant sur Ubuntu 12.04 64bits.

2.2.1 Résolution partielle des LMIs de la caractérisation par matrices (D,G)

Parfois, l'utilisation d'une technique de descente de gradient pour calculer $\bar{\mu}_\Delta$ donne une valeur trop conservative. A l'opposé, le conservatisme est souvent amélioré si un solveur LMI est employé, au détriment du temps de calcul. Ceci est illustré sur le tableau 2.2, reprenant une partie des benchmarks présentés dans le tableau 1.1. $\bar{\mu}_\Delta^{\text{LMI}}$ correspond à la borne supérieure obtenue en utilisant un solveur LMI, alors que $\bar{\mu}_\Delta^{\text{GD}}$ est évaluée par une technique de descente de gradient. On observe bien sur cette liste de benchmarks que l'écart entre les deux méthodes est grand et que le temps de calcul de $\bar{\mu}_\Delta^{\text{LMI}}$ est très important lorsque le nombre d'incertitudes, et surtout le nombre de répétitions, est élevé.

Afin d'améliorer la complexité numérique en présence d'incertitudes Δ de grande taille, l'idée est d'initialiser les matrices D et G en utilisant une approche par

Benchmark	$\bar{\mu}_{\Delta}^{\text{LMI}}$		$\bar{\mu}_{\Delta}^{\text{GD}}$		
	Valeur	Temps de calcul	Valeur	Écart avec $\bar{\mu}_{\Delta}^{\text{LMI}}$	Temps de calcul
3	1.1332	26s	1.2893	12.56%	5s
4	1.0156	122s	1.6912	66.52%	4s
7	46.5977	63s	56.1098	20.41%	9s
8	1.2144	306s	1.4618	20.47%	6s
9	2.7232	3546s	4.4489	63.37%	36s
11	0.3839	2555s	1.2687	230.48%	22s

TABLEAU 2.2 – Liste de benchmarks présentant un écart important entre $\bar{\mu}_{\Delta}^{\text{LMI}}$ et $\bar{\mu}_{\Delta}^{\text{GD}}$, ainsi qu'un temps de calcul élevé pour $\bar{\mu}_{\Delta}^{\text{LMI}}$.

descente de gradient, puis d'en optimiser certaines parties seulement en utilisant un solveur LMI. Le nombre de variables de décision n_{dec} est choisi par l'utilisateur, permettant un compromis entre temps de calcul et précision de la borne obtenue $\bar{\mu}_{\Delta}^{\text{pLMI}}$. Les μ -sensibilités sont calculées afin de déterminer les incertitudes ayant le plus d'influence sur la valeur de la borne supérieure, et donc les blocs des matrices D et G à optimiser. On peut alors s'attendre à :

- faire tendre la valeur de la borne supérieure obtenue vers $\bar{\mu}_{\Delta}^{\text{LMI}}$,
- obtenir un temps de calcul modérément supérieur à celui nécessaire pour évaluer $\bar{\mu}_{\Delta}^{\text{GD}}$.

L'algorithme 2 est proposé dans ce contexte.

Algorithme 2: Calcul d'une borne supérieure de $\bar{\mu}_\Delta$ par optimisation LMI partielle des matrices D et G

Entrées : $M(s), \Delta, n_{dec}$

Sorties : $\bar{\mu}_\Delta^{\text{pLMI}}$

- 1 Initialiser $\mathcal{I} = \{\mathbb{R}^+\}$, la liste des intervalles fréquentiels sur laquelle la borne supérieure doit être évaluée ;
 - 2 Initialiser $\beta_{max} = 0$;
 - tant que** $\mathcal{I} \neq \emptyset$
 - 3 Choisir un intervalle $I \in \mathcal{I}$ et une fréquence $\omega \in I$. Définir $M = M(j\omega)$;
 - 4 Calculer D^{gd} et G^{gd} minimisant β dans (1.21)-(1.23), par une approche de descente de gradient ;
 - 5 Calculer les μ -sensibilités pour chaque $\delta_j \in \Delta$ en utilisant l'algorithme 1. Trier ces incertitudes selon un ordre décroissant des valeurs des μ -sensibilités. Calculer les matrices M, D^{gd} et G^{gd} correspondantes par manipulations de lignes et de colonnes ;
 - 6 Définir $D^{lmi} = D^{gd}$ et $G^{lmi} = G^{gd}$;
 - 7 Minimiser β dans (1.21) en optimisant les premiers n_{dec} éléments de D^{lmi} et G^{lmi} , et en laissant les autres inchangés ;
 - 8 Définir $\beta_{max} = \max(\beta, \beta_{max})$;
 - 9 Exécuter la technique de balayage de fréquences de [Roos et Biannic 2010] – rappelée brièvement au chapitre 1 – afin de déterminer le plus grand intervalle fréquentiel I_v pour lequel la borne supérieure β_{max} et les matrices D^{lmi} et G^{lmi} sont valables ;
 - 10 Mettre à jour $\mathcal{I} \leftarrow \mathcal{I} \setminus I_v$;
 - 11 $\bar{\mu}_\Delta^{\text{pLMI}} = \beta_{max}$ est une borne supérieure de μ_Δ qui vérifie $\bar{\mu}_\Delta^{\text{LMI}} \leq \bar{\mu}_\Delta^{\text{pLMI}} \leq \bar{\mu}_\Delta^{\text{GD}}$.
-

Remarque 2.6

Le nombre de variables de décision n_{dec} doit être choisi avec précaution. Une valeur petite améliorera sensiblement le temps de calcul mais ne permettra pas de réduire suffisamment la valeur de la borne supérieure par rapport à $\bar{\mu}_\Delta^{\text{GD}}$. Une valeur trop élevée risque à l'inverse de considérablement augmenter le temps de calcul même si la borne obtenue sera proche de $\bar{\mu}_\Delta^{\text{LMI}}$. Cependant un compromis est facilement déterminé, permettant d'obtenir de bons résultats dans quasiment tous les cas.

Les résultats sont présentés dans le tableau 2.3, pour les benchmarks sélectionnés dans le tableau 2.2. Le nombre de variables de décision est fixé à $n_{dec} = 80$, afin de garder un temps de calcul raisonnable. L'écart entre $\bar{\mu}_\Delta^{\text{LMI}}$ et $\bar{\mu}_\Delta^{\text{pLMI}}$ est faible (troisième colonne), en comparaison avec l'écart initial entre $\bar{\mu}_\Delta^{\text{LMI}}$ et $\bar{\mu}_\Delta^{\text{GD}}$ (cinquième

colonne). Le temps de calcul est fortement diminué, tendant vers celui nécessaire pour évaluer $\bar{\mu}_{\Delta}^{\text{GD}}$. Ces résultats prouvent bien la capacité de la méthode à calculer une borne supérieure proche de $\bar{\mu}_{\Delta}^{\text{LMI}}$ avec un temps de calcul assez proche de l'approche par descente de gradient.

Benchmark	$\bar{\mu}_{\Delta}^{\text{LMI}}$	$\bar{\mu}_{\Delta}^{\text{pLMI}}$ (algorithme 2)		$\bar{\mu}_{\Delta}^{\text{GD}}$	
	Temps CPU	Écart avec $\bar{\mu}_{\Delta}^{\text{LMI}}$	Temps CPU	Écart avec $\bar{\mu}_{\Delta}^{\text{LMI}}$	Temps CPU
3	26s	0.43%	17s	12.56%	5s
4	122s	14.60%	30s	66.52%	4s
7	63s	0.01%	20s	20.41%	9s
8	306s	0.57%	30s	20.47%	6s
9	3546s	26.58%	100s	63.37%	36s
11	2555s	97.34%	49s	230.48%	22s

TABLEAU 2.3 – Résultats de l'utilisation de la résolution partielle de LMIs avec aide des μ -sensibilités (algorithme 2)

Remarque 2.7

Pour le Benchmark N° 11, l'écart semble encore assez élevé. Cependant il faut remarquer que la valeur de $\bar{\mu}_{\Delta}^{\text{pLMI}} < 1$ permet de conclure sur la stabilité robuste du système en un temps très court, alors que le fort conservatisme de $\bar{\mu}_{\Delta}^{\text{GD}} > 1$ ne le permet pas.

2.2.2 Nouvel algorithme de branch-&-bound

Lorsque le conservatisme entre les bornes supérieure et inférieure reste élevé, même malgré l'utilisation d'un solveur LMI, la technique de branch-&-bound peut être très appropriée. Elle a été introduite par [Newlin et Young 1997] dans le cadre de la μ -analyse et appliquée à l'approche par matrices (D,G). Le branch-&-bound est une méthode générale applicable à des problèmes d'optimisation, où l'on cherche à calculer des bornes supérieure et inférieure qui dépendent du domaine d'optimisation : plus le domaine est petit, plus l'écart entre les bornes diminue. La technique consiste donc à partitionner le domaine en des sous-ensembles de plus en plus petits jusqu'à ce que l'écart entre les bornes soit inférieur à une tolérance prédéfinie.

Clairement, le problème du branch-&-bound se situe au niveau de son temps de calcul qui peut devenir vite prohibitif si le nombre d'incertitudes s'avère trop élevé, et si la tolérance définie est trop petite. En effet, à chaque itération, le domaine d'incertitudes est divisée en deux nouveaux sous-domaines, ce qui en fait une technique à temps exponentiel. Elle est donc à utiliser très ponctuellement et ne consiste pas en une approche systématique pour calculer une borne supérieure de μ_{Δ} , du

moins dans sa version classique. Il est proposé dans cette sous-section d'améliorer le temps de convergence, grâce à l'utilisation des μ -sensibilités. Le principe est simple : plutôt que de subdiviser à chaque itération le domaine d'incertitudes – qui dans le cadre de la μ -analyse est un hyperrectangle – suivant sa plus grande dimension, la dimension possédant la μ -sensibilité la plus forte est considérée. Dans ce contexte, un nouvel algorithme 3 de branch-&-bound est proposé, qui nécessite seulement à l'utilisateur de définir la tolérance η correspondant à l'écart final entre les bornes supérieure et inférieure de μ_Δ .

Algorithme 3: Calcul d'une borne supérieure de μ_Δ par branch-&-bound avec utilisation des μ -sensibilités

Entrées : $M(s)$, Δ , η

Sorties : $\bar{\mu}_\Delta^{\text{bb}}, \underline{\mu}_\Delta^{\text{bb}}$

- 1 Calculer une borne inférieure de μ_Δ , dénotée $\underline{\mu}_\Delta^{\text{bb}}$, par l'utilisation des algorithmes cités dans les sections 1.2.2 et 1.3 ;
 - 2 Définir $\bar{\mu}_\Delta^{\text{test}} = (1 + \eta)\underline{\mu}_\Delta^{\text{bb}}$ la valeur candidate de la borne supérieure de μ_Δ ;
 - 3 Définir $\mathcal{E} = \{(\mathcal{S}_0, \Omega_0)\}$ comme la liste initiale des éléments à tester, avec $\mathcal{S}_0 = \frac{1}{\bar{\mu}_\Delta^{\text{test}}} \mathcal{B}_\Delta$ et $\Omega_0 = \mathbb{R}^+$;
 - tant que** $\mathcal{E} \neq \emptyset$
 - 4 Choisir un élément (\mathcal{S}, Ω) de \mathcal{E} ;
 - 5 Calculer $\tilde{M}(s)$ à partir de $M(s)$ tel que les interconnexions $(M(s), \Delta \in \mathcal{S})$ et $(\tilde{M}(s), \Delta \in \mathcal{B}_\Delta)$ soient équivalentes ;
 - 6 Déterminer les fréquences $\Omega_{ok} \subset \Omega$ pour lesquelles $\bar{\mu}_\Delta < 1$ pour l'interconnexion normalisée $(\tilde{M}(s), \Delta \in \mathcal{B}_\Delta)$ calculée lors de l'étape précédente ;
 - si** $\Omega_{ok} \neq \Omega$ **alors**
 - 7 Calculer les μ -sensibilités pour une fréquence $\omega \in \Omega_{nok} = \Omega \setminus \Omega_{ok}$, en utilisant l'algorithme 1 ;
 - 8 Partitionner \mathcal{S} selon la direction de μ -sensibilité maximale afin d'obtenir deux nouveaux sous-domaines \mathcal{S}_1 et \mathcal{S}_2 ;
 - 9 Ajouter $(\mathcal{S}_1, \Omega_{nok})$ et $(\mathcal{S}_2, \Omega_{nok})$ à \mathcal{E} ;
 - sinon**
 - Enlever (\mathcal{S}, Ω) de \mathcal{E} ;
 - 10 $\bar{\mu}_\Delta^{\text{bb}} = \bar{\mu}_\Delta^{\text{test}}$ est une borne supérieure de μ_Δ .
-

Remarque 2.8

Toutes les 10 ou 20 itérations, une borne inférieure de μ_Δ est réévaluée. Si $\underline{\mu}_\Delta^{\text{bb}}$ augmente, alors $\bar{\mu}_\Delta^{\text{test}}$ est mis à jour comme décrit à l'étape 2, et le domaine \mathcal{S} associé à chaque élément de \mathcal{E} est également ajusté en fonction. Ceci permet d'assurer la convergence de l'algorithme dans le cas où l'écart entre la valeur initiale de $\underline{\mu}_\Delta^{\text{bb}}$ et celle de μ_Δ est supérieure à η .

Remarque 2.9

Le paramètre η , correspondant à l'écart désiré entre les bornes, permet un compromis entre la précision désirée et un temps de calcul raisonnable.

Les résultats sont présentés dans le tableau 2.4, pour une liste de benchmarks issus du tableau 1.1, dont le conservatisme est initialement très élevé, même avec l'utilisation d'un solveur LMI pour calculer les matrices D et G . Le branch-&-bound est donc une solution à plébisciter dans ces cas-ci, mais il est important de noter que l'utilisation de la technique de branch-&-bound classique basée sur [Newlin et Young 1997] entraîne un temps de calcul très important, comme le montre la 3^{me} colonne du tableau 2.4. Avec l'utilisation de l'algorithme 3, le temps nécessaire pour que l'écart entre les bornes soit inférieur à la tolérance η est bien plus court. Le temps de convergence est donc fortement diminué grâce à l'utilisation des μ -sensibilités. Il est intéressant de remarquer notamment que pour le benchmark 9, le branch-&-bound classique ne permet pas de converger en un temps raisonnable (inférieur à 24 heures) lorsqu'un écart entre les bornes inférieure et supérieure de μ_Δ de 10% est souhaité, alors que l'algorithme 3 converge en 900 secondes. Tous ces résultats démontrent la capacité de ce nouvel algorithme à diminuer drastiquement le conservatisme de la borne supérieure de μ_Δ tout en assurant un temps de convergence raisonnable.

Benchmark	η désiré	Branch & bound classique	Algorithme 3
		Temps CPU	Temps CPU
3	2%	1787s	104s
5	5%	348s	180s
6	5%	688s	481s
9	10%	∞	901s
	85%	280s	10s
10	10%	36269s	1485s

TABLEAU 2.4 – Résultats de l'utilisation du branch-&-bound sur une liste de benchmarks, avec ou sans l'utilisation des μ -sensibilités.

2.3 UNE BORNE SUPÉRIEURE DE μ MOINS CONSERVATIVE PAR L'APPROCHE PAR MULTIPLIEURS

Dans cette section, il est proposé d'utiliser une autre approche pour le calcul de la borne supérieure de μ_Δ , moins conservative que celle par matrices (D,G) décrite à la section précédente. L'approche dite par multiplieurs, introduite par [Fu et Barabanov 1997], est en fait une généralisation de la formulation par matrices (D,G), et fournit donc en théorie une borne supérieure de μ_Δ moins conservative. Cepen-

dant, deux problèmes majeurs limitent l'utilisation de cette technique notamment pour des applications réalistes :

- la complexité numérique augmente exponentiellement avec le nombre d'incertitudes,
- la borne supérieure de μ_Δ fournie est seulement calculée à une fréquence donnée, et il n'existe pas de méthodes pour valider cette borne sur \mathbb{R}^+ . Il est alors nécessaire d'utiliser un maillage fréquentiel, avec tous les inconvénients évoqués à la section 1.2.3.

Ces deux principaux problèmes sont traités dans cette section, afin de rendre l'approche par multipliers plus attrayante pour des applications réalistes.

2.3.1 Rappels de l'approche

Définition

Une fois encore, on considère l'interconnexion de la figure 1.2, où $M(s) \in \mathcal{RH}_\infty^{n \times n}$ représente un système nominal bouclé LTI, stable, et $\Delta(s) = \Delta$ est une matrice diagonale par blocs regroupant les incertitudes de modélisation. Sans perte de généralité, on se focalise ici uniquement sur le cas particulier entraînant le plus de conservatisme, celui où les incertitudes sont uniquement paramétriques réelles. Les incertitudes complexes tendant à réduire le conservatisme, elles ne sont pas considérées ici. La matrice Δ a donc la structure suivante :

$$\Delta = \text{diag}(\delta_1 I_{k_1}, \dots, \delta_r I_{k_r}) \quad (2.17)$$

où les δ_i sont des scalaires réels répétés k_i -fois, et $\sum_{i=1}^r k_i = n$. On rappelle que Δ contient l'ensemble des matrices de taille $n \times n$ possédant la même structure que Δ dans (2.17), et on définit deux nouvelles notations pour cette section :

$$\mathcal{B}_\Delta = \{\Delta \in \Delta : |\delta_i| \leq 1\} \quad (2.18)$$

$$\bar{\mathcal{B}}_\Delta = \{\Delta \in \Delta : |\delta_i| = \pm 1\} \quad (2.19)$$

On introduit tout d'abord le lemme suivant, qui constitue la base de l'approche par multipliers :

Lemme 2.2

Une famille de matrices \mathcal{A} est non-singulière s'il existe une autre matrice C , appelée multiplieur, telle que $\text{He } CA < 0$ pour tout $A \in \mathcal{A}$, avec $\text{He } X$ la partie hermitienne de la matrice X .

Démonstration : Par contraposée, supposons qu'il existe $A \in \mathcal{A}$ telle que A est sin-

gulière. Alors il existe un vecteur $v \neq 0$, tel que $Av = 0$. On obtient donc que $v^*(CA + A^*C^*)v = 0$, ce qui signifie que $\text{He } CA$ n'est pas définie négative. ■

Le théorème suivant découle de l'utilisation du lemme 2.2 et de la définition 1.2.

Théorème 2.2

Soit α un scalaire positif, et $M = M(j\omega)$ la réponse fréquentielle de $M(s)$ à une fréquence donnée ω . S'il existe un multiplieur $C \in \mathbb{C}^{n \times n}$ tel que :

$$\text{He } C^*(I_n - \Delta M) < 0 \quad (2.20)$$

pour tout $\Delta \in \frac{1}{\alpha}\mathcal{B}_\Delta$, alors $\mu_\Delta(M) \leq \alpha$.

Puisque l'équation (2.20) est linéaire en δ_i , l'ensemble \mathcal{B}_Δ peut être remplacé de façon équivalente par l'ensemble $\tilde{\mathcal{B}}_\Delta$. Afin d'évaluer une borne supérieure de $\mu_\Delta(M)$, le but est donc de calculer l'infimum $\alpha > 0$ tel que l'équation (2.20) est valide pour tout $\Delta \in \frac{1}{\alpha}\tilde{\mathcal{B}}_\Delta$. Ceci peut être formulé comme un problème aux valeurs propres généralisé (en anglais, GEVP) :

$$\begin{aligned} v_\Delta(M) &= \min_{\alpha \in \mathbb{R}^+, C \in \mathbb{C}^{n \times n}} \alpha \\ \text{tel que } &\text{He } C^*(\alpha I_n - \Delta M) < 0, \quad \forall \Delta \in \tilde{\mathcal{B}}_\Delta \end{aligned} \quad (2.21)$$

où $v_\Delta(M)$ est appelé la borne supérieure de $\mu_\Delta(M)$ basée sur l'approche par multiplieurs. La résolution du GEVP (2.21) demande de résoudre un ensemble de 2^r LMIs, correspondant à chaque sommet de l'hypercube $\tilde{\mathcal{B}}_\Delta$. Il est donc important de souligner que l'approche par multiplieurs est de complexité exponentielle. Le multiplieur C , complexe, non-structuré, et de taille $n \times n$, est quant à lui constitué de $2n^2$ variables de décision. Cependant, il existe de nombreux cas pratiques où la matrice M n'est pas de rang plein, de sorte qu'on peut écrire :

$$M = AB^* \quad (2.22)$$

avec $A, B \in \mathbb{C}^{n \times q}$ et $q = \text{rang}(M) < n$. On peut alors remplacer la formulation (2.20) de façon équivalente par :

$$\text{He } \hat{C}^*(I_q - B^*\Delta A) < 0 \quad (2.23)$$

où $\hat{C} \in \mathbb{C}^{q \times q}$. L'utilisation de l'équation (2.23) est bien plus avantageuse puisque la taille du multiplieur à optimiser passe de $n \times n$ à $q \times q$.

Du conservatisme de l'approche par multiplieurs

Le théorème suivant vise à démontrer que la borne supérieure de $\mu_{\Delta}(M)$ obtenue par l'approche par multiplieurs, notée $\nu_{\Delta}(M)$, est moins conservatrice que celle issue de l'approche par matrices (D,G), notée $\bar{\mu}_{\Delta}(M)$.

Théorème 2.3

Pour toutes les structures Δ et matrices M , on a :

$$\nu_{\Delta}(M) \leq \bar{\mu}_{\Delta}(M) \quad (2.24)$$

Afin de démontrer ce théorème, on introduit dans un premier temps le lemme suivant, issu de [Gahinet et Apkarian 1994] et aussi connu sous le nom du lemme de projection :

Lemme 2.3

Soient $U, V \in \mathbb{C}^{n \times n}$ et $Q = Q^* \in \mathbb{C}^{n \times n}$. Il existe $T \in \mathbb{C}^{n \times n}$ tel que :

$$Q + U^*TV + V^*T^*U < 0 \quad (2.25)$$

si et seulement si les deux conditions suivantes sont satisfaites :

$$U_{\perp}^*QU_{\perp} < 0, \quad U_{\perp}^*QV_{\perp} < 0 \quad (2.26)$$

où U_{\perp} et V_{\perp} sont n'importe quelles matrices dont les colonnes forment des bases des noyaux de U et V , respectivement.

On peut donc donner la démonstration du théorème 2.3, inspirée de [Fu et Barabanov 1997].

Démonstration du théorème 2.3 : Soit $\Phi_{\beta}(D, G) = M^*DM + j(GM - M^*G) - \beta^2D$.

Deux étapes sont nécessaires afin de démontrer que $\Phi_{\beta}(D, G) < 0$ implique que $\text{He } C^*(I_n - \Delta M) < 0$ pour tout $\Delta \in (1/\beta)\bar{\mathcal{B}}_{\Delta}$, prouvant ainsi que $\nu_{\Delta}(M) \leq \bar{\mu}_{\Delta}(M)$.

Étape 1) Montrons que $\Phi_{\beta}(D, G) < 0$ si et seulement si il existe $C \in \mathbb{C}^{n \times n}$ tel que :

$$W(C, D, G) = \begin{bmatrix} C + C^* + M^*DM & C^* + jM^*G \\ C - jGM & -\beta^2D \end{bmatrix} < 0 \quad (2.27)$$

Pour ce faire, réécrivons tout d'abord (2.27) comme suit :

$$W(C, D, G) = \begin{bmatrix} M^*DM & jM^*G \\ -jGM & -\beta^2D \end{bmatrix} + \begin{bmatrix} I_n \\ I_n \end{bmatrix} C \begin{bmatrix} I_n & 0 \end{bmatrix} + \begin{bmatrix} I_n \\ 0 \end{bmatrix} C^* \begin{bmatrix} I_n & I_n \end{bmatrix} < 0 \quad (2.28)$$

Il est facile de remarquer que $\begin{bmatrix} 0 \\ I_n \end{bmatrix}$ et $\begin{bmatrix} -I_n \\ I_n \end{bmatrix}$ forment des bases des noyaux de

$\begin{bmatrix} I_n & 0 \end{bmatrix}$ et $\begin{bmatrix} I_n & I_n \end{bmatrix}$ respectivement. En utilisant le lemme 2.3, (2.28) est vérifié si et seulement si $-\beta^2 D < 0$ et :

$$M^*DM + j(GM - M^*G) - \beta^2 D < 0 \quad (2.29)$$

La première condition est redondante et (2.29) correspond à $\Phi_\beta(D, G) < 0$.

Étape 2) Soit $Z = \begin{bmatrix} I_n \\ -\Delta M \end{bmatrix}$. Il apparaît clairement que :

$$Z^*WZ = C^*(I_n - \Delta M) + (I_n - \Delta M)^*C + M^*D^{1/2}(I - \beta^2\Delta^2)D^{1/2}M \quad (2.30)$$

puisque $D\Delta = \Delta D$ et $G\Delta = \Delta G$. Ainsi, $\Phi_\beta(D, G) < 0 \Leftrightarrow W(C, D, G) < 0$ implique que $He C^*(I_n - \Delta M) < 0$ pour tout $\Delta \in (1/\beta)\bar{\mathcal{B}}_\Delta$, ce qui démontre bien que $\nu_\Delta(M) \leq \bar{\mu}_\Delta(M)$. ■

Maintenant que les principales notions de l'approche par multiplieurs pour calculer une borne supérieure de μ_Δ à une fréquence donnée ont été rappelées, la sous-section suivante se focalise sur la réduction du temps de calcul de cette borne, en utilisant – comme pour les solutions proposées pour l'approche par matrices (D,G) – les μ -sensibilités.

2.3.2 μ -sensibilités et réduction du temps de calcul

La résolution du problème aux valeurs propres généralisé (2.21) nécessite de considérer 2^r LMIs. Ainsi, l'évaluation de $\nu_\Delta(M)$ peut être extrêmement coûteuse si le nombre d'incertitudes est élevé. Dans l'esprit de ce qui a été proposé pour la résolution des LMIs dans l'approche par matrices (D,G), l'idée ici est de résoudre (2.21) uniquement pour un nombre limité $r_1 < r$ d'incertitudes, et d'utiliser la formulation basée sur les matrices (D,G) pour les $r - r_1$ restantes. La borne supérieure obtenue, notée $\nu_\Delta^p(M)$, sera nécessairement comprise entre $\nu_\Delta(M)$ et $\bar{\mu}_\Delta(M)$, soit légèrement plus conservative mais bien plus attrayante en terme de temps de calcul que $\nu_\Delta(M)$. Cette solution est déjà envisagée dans [Fu et Barabanov 1997], mais sans aucune indication sur comment choisir efficacement les r_1 blocs d'incertitudes de Δ de telle sorte que l'augmentation du conservatisme soit la plus faible possible.

On se propose ici d'utiliser les μ -sensibilités introduites à la section 2.1, afin de sélectionner les r_1 blocs les plus critiques de Δ . M et Δ sont réarrangés de façon à ce que $\Delta = \text{diag}(\Delta_1, \Delta_2)$ avec $\Delta_1 \in \Delta_1 \subset \mathbb{R}^{p \times p}$ contenant les r_1 blocs d'incertitudes (et $p = \sum_{i=1}^{r_1} k_i$), et Δ_2 les autres $r - r_1$ blocs. Dans le cas où M est de rang réduit,

on peut également décomposer M de la façon suivante :

$$M = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \begin{bmatrix} B_1^* & B_2^* \end{bmatrix} \quad (2.31)$$

avec $A_1, B_1 \in \mathbb{C}^{p \times q}$, $A_2, B_2 \in \mathbb{C}^{(n-p) \times q}$, et $q = \text{rang}(M)$. Ainsi, le problème suivant doit être résolu :

$$\begin{aligned} \nu_{\Delta}^p(M) &= \min_{\alpha \in \mathbb{R}^+, \hat{C} \in \mathbb{C}^{q \times q}, G \in \mathcal{G}_{n-p}, D \in \mathcal{D}_{n-p}} \alpha \\ \text{tel que } &\begin{bmatrix} L_{11} & \hat{C}^* B_2^* + j A_2^* G \\ * & -\alpha^2 D \end{bmatrix} < 0, \quad \forall \Delta_1 \in \frac{1}{\alpha} \bar{\mathcal{B}}_{\Delta_1} \\ &\text{avec } L_{11} = 2\text{He } \hat{C}^* (I_q - B_1^* \Delta_1 A_1) + A_2^* D A_2 \end{aligned} \quad (2.32)$$

Il n'implique plus la résolution de 2^r LMIs, mais uniquement de 2^{r_1} où $r_1 < r$ est fixé par l'utilisateur. Les μ -sensibilités permettent de plus de sélectionner les incertitudes les plus pertinentes pour ne pas augmenter de façon trop significative le conservatisme par rapport à $\nu_{\Delta}(M)$.

Il est également important de noter qu'une solution du problème (2.32) est également une solution du problème initial (2.23), comme le prouve le théorème suivant.

Théorème 2.4

Soient $\alpha > 0$ et $\hat{C} \in \mathbb{C}^{q \times q}$ solutions du problème (2.32). Alors $\text{He } \hat{C}^* (I_q - B^* \Delta A) < 0$ pour tout $\Delta \in (1/\alpha) \bar{\mathcal{B}}_{\Delta}$.

Démonstration du théorème 2.4 : Soient $\alpha > 0$, $\hat{C} \in \mathbb{C}^{q \times q}$, $D \in \mathcal{D}_{n-p}$ et $G \in \mathcal{G}_{n-p}$ tels que :

$$L(\hat{C}, D, G) = \begin{bmatrix} L_{11} & \hat{C}^* B_2^* + j A_2^* G \\ * & -\alpha^2 D \end{bmatrix} < 0, \quad \forall \Delta_1 \in \frac{1}{\alpha} \bar{\mathcal{B}}_{\Delta_1}$$

Soit $Y = \begin{bmatrix} I_q \\ -\Delta_2 A_2 \end{bmatrix}$. De simples calculs permettent d'obtenir que :

$$Y^* L Y = \hat{C}^* (I_q - A^* \Delta B)^* + (I_q - A^* \Delta B) \hat{C} + A_2^* D_2^{1/2} (I - \alpha^2 \Delta_2^2) D_2^{1/2} A_2 \quad (2.33)$$

Ainsi, $\text{He } \hat{C}^* (I_q - B^* \Delta A) < 0$ pour tout $\Delta \in (1/\alpha) \bar{\mathcal{B}}_{\Delta}$. ■

Une méthode a donc été proposée pour réduire la complexité numérique du calcul d'une borne supérieure de μ à une fréquence donnée à l'aide de l'approche par multiplieurs. Elle utilise d'une part les μ -sensibilités pour n'appliquer la caracté-

risation la moins conservative qu'aux incertitudes les plus pertinentes, réduisant ainsi le nombre de LMIs, et d'autre part le rang souvent réduit en pratique de la matrice M , réduisant cette fois le nombre de variables de décision du multiplieur.

Cependant il reste le problème de l'évaluation de $\nu_\Delta(M)$ sur \mathbb{R}^+ tout entier sans utiliser un maillage fréquentiel. La sous-section suivante propose une solution, basée sur une approche hamiltonienne, comme ce qui a été fait pour l'approche par matrices (D,G) (voir sous-section 1.2.3 du chapitre 1).

2.3.3 Validation des multiplieurs sur de larges intervalles fréquentiels

Le calcul d'une marge de robustesse sur \mathbb{R}^+ n'est pas évident. L'utilisation d'un maillage fin peut impliquer un temps de calcul trop important sans avoir la certitude qu'une fréquence critique n'est pas manquée, conduisant à une surestimation non acceptable de la marge de robustesse k_{max} .

Cette sous-section présente une méthode qui détermine pour quels intervalles de fréquences le multiplieur C (ou \hat{C}) et $\nu_\Delta(M)$ (ou $\nu_\Delta^p(M)$) évalués à une fréquence donnée restent valides. Les résultats sont basés sur le théorème suivant.

Théorème 2.5

Soient (A_M, B_M, C_M, D_M) les matrices de la représentation d'état de $M(s)$ avec A_M Hurwitz. Soient $\alpha > 0$ et $C \in \mathbb{C}^{n \times n}$ tels que, pour une fréquence donnée ω et une matrice $\Delta \in \frac{1}{\alpha} \tilde{\mathcal{B}}_\Delta$:

$$\text{He } C^*(I_n - \Delta M(j\omega)) < 0 \quad (2.34)$$

Soit $X = 2\text{He } C^*(I_n - \Delta D_M)$. Si X est non-singulière, alors $\text{He } C^*(I_n - \Delta M(j\tilde{\omega}))$ est singulière si et seulement si $j\tilde{\omega}$ est une valeur propre de la matrice Hamiltonienne suivante :

$$H = \begin{bmatrix} A_M & 0 \\ 0 & -A_M^* \end{bmatrix} + \begin{bmatrix} B_M \\ C_M^* \Delta C \end{bmatrix} X^{-1} \begin{bmatrix} C^* \Delta C_M & -B_M^* \end{bmatrix} \quad (2.35)$$

Le théorème 2.5 peut être utilisé afin de calculer les intervalles fréquentiels \mathcal{I}_v pour lesquels l'équation (2.34) est satisfaite pour la matrice d'incertitudes Δ et le multiplieur C considérés. Soient $j\tilde{\omega}_1, \dots, j\tilde{\omega}_N$ les valeurs propres purement imaginaires de H telles que $0 < \tilde{\omega}_1 < \dots < \tilde{\omega}_N$. Il est évident que $[0 \ \tilde{\omega}_1] \cup [\tilde{\omega}_1 \ \tilde{\omega}_2] \cup \dots \cup [\tilde{\omega}_{N-1} \ \tilde{\omega}_N] \cup [\tilde{\omega}_N \ +\infty[= \mathbb{R}^+$. Ainsi il suffit de prendre une fréquence ω_t au sein de chacun de ces $N + 1$ intervalles et de tester si $\text{He } C^*(I_n - \Delta M(j\omega_t)) < 0$. \mathcal{I}_v est composé de tous les intervalles pour lesquels cette condition est satisfaite.

Remarque 2.10

Dans le théorème 2.5, la condition de non-singularité sur la matrice X n'est pas tant restrictive en pratique. En effet, il arrive souvent que la matrice de transmission directe

D_M soit nulle lorsqu'on considère des applications réalistes. Dans ce cas, la matrice X est toujours non-singulière, puisqu'on a alors $X = 2\text{He } C^ < 0$. De plus, pour tous les benchmarks qui ont été étudiés afin de valider la méthode où la matrice D_M n'était pas identiquement nulle, cette condition de non-singularité était satisfaite.*

Afin de démontrer le théorème 2.5, le lemme suivant et sa preuve sont tout d'abord introduits.

Lemme 2.4

Soit $P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}$ une matrice avec P_{11} et P_{22} inversibles. Les égalités suivantes sont satisfaites :

$$\begin{aligned} \det(P) &= \det(P_{11})\det(P_{22} - P_{21}P_{11}^{-1}P_{12}) \\ &= \det(P_{22})\det(P_{11} - P_{12}P_{22}^{-1}P_{21}) \end{aligned} \quad (2.36)$$

Démonstration du lemme 2.4 : On a les égalités suivantes :

$$P \begin{bmatrix} I & -P_{11}^{-1}P_{12} \\ 0 & I \end{bmatrix} = \begin{bmatrix} P_{11} & 0 \\ P_{21} & P_{22} - P_{21}P_{11}^{-1}P_{12} \end{bmatrix} \quad (2.37)$$

En prenant le déterminant de (2.37), on obtient :

$$\begin{aligned} \det(P) &= \det \left(\begin{bmatrix} P_{11} & 0 \\ P_{21} & P_{22} - P_{21}P_{11}^{-1}P_{12} \end{bmatrix} \right) \\ &= \det(P_{11})\det(P_{22} - P_{21}P_{11}^{-1}P_{12}) \end{aligned} \quad (2.38)$$

La seconde égalité est obtenue de la même manière en post-multipliant P par $\begin{bmatrix} I & 0 \\ -P_{22}^{-1}P_{21} & I \end{bmatrix}$. ■

La démonstration du théorème 2.5 peut maintenant être introduite.

Démonstration du théorème 2.5 : Soient $\alpha > 0$ et $C \in \mathbb{C}^{n \times n}$ tels que $F(j\omega) = 2\text{He } C^*(I_n - \Delta M(j\omega)) < 0$. On a alors l'égalité suivante :

$$F(j\omega) = \begin{bmatrix} (j\omega I - A_M)^{-1}B_M \\ I \end{bmatrix}^* \begin{bmatrix} 0 & -C_M^* \Delta C \\ -C^* \Delta C_M & X \end{bmatrix} \begin{bmatrix} (j\omega I - A_M)^{-1}B_M \\ I \end{bmatrix} \quad (2.39)$$

avec $X = 2\text{He } C^*(I_n - \Delta D_M)$. Soit alors la matrice suivante :

$$Q = \begin{bmatrix} j\tilde{\omega}I - A_M & 0 & -B_M \\ 0 & j\tilde{\omega}I + A_M^* & -C_M^* \Delta C \\ -C^* \Delta C_M & B_M^* & X \end{bmatrix} \quad (2.40)$$

Puisque A_M est Hurwitz, $j\tilde{\omega}I - A_M$ et $j\tilde{\omega}I + A_M^*$ sont non-singulières. Par hypothèse, X est également non-singulière. L'application du lemme 2.4 implique finalement que :

$$\begin{aligned} \det(Q) &= \det(j\tilde{\omega}I - H) \underbrace{\det(X)}_{\neq 0} \\ &= \underbrace{\det(j\tilde{\omega}I + A_M^*)}_{\neq 0} \det(F(j\tilde{\omega})) \underbrace{\det(j\tilde{\omega}I - A_M)}_{\neq 0} \end{aligned}$$

après quelques manipulations matricielles. Ainsi, $F(j\tilde{\omega})$ est singulière si et seulement si $j\tilde{\omega}$ est une valeur propre de H . ■

Il est important de noter que le théorème 2.5 est seulement valide dans le cas où le multiplieur C est de rang plein, soit de taille $n \times n$. Or lors des sous-sections précédentes, il a été montré qu'il était possible de réduire la taille du multiplieur en prenant en considération une dégénérescence éventuelle du rang de M , faisant passer ainsi la taille du multiplieur C de $n \times n$ au multiplieur \hat{C} de taille $q \times q$, où $q = \text{rang}(M) < n$. Ceci constitue un avantage non négligeable en termes de complexité numérique qu'il serait dommage de laisser de côté. C'est pourquoi le théorème 2.6 est proposé ci-après afin de calculer un multiplieur de rang plein $C \in \mathbb{C}^{n \times n}$ valide, à partir du multiplieur $\hat{C} \in \mathbb{C}^{q \times q}$.

Théorème 2.6

Soit $M = M(j\omega)$ la réponse fréquentielle de $M(s)$ à une fréquence donnée ω telle que $M = AB^*$, avec $A, B \in \mathbb{C}^{n \times q}$ et $q = \text{rang}(M)$. Soient $\hat{C} \in \mathbb{C}^{q \times q}$ et $\alpha > 0$ tels que :

$$\text{He } \hat{C}^*(I_q - B^* \Delta A) < 0, \quad \forall \Delta \in \frac{1}{\alpha} \bar{\mathcal{B}}_\Delta \quad (2.41)$$

Soit $C = B\hat{C}B^* - \epsilon EE^* \in \mathbb{C}^{n \times n}$, avec $\epsilon > 0$ assez petit, et $E \in \mathbb{C}^{n \times (n-q)}$ tel que $\begin{bmatrix} B & E \end{bmatrix}$ soit de rang plein. Alors :

$$\text{He } C^*(I_n - \Delta M) < 0, \quad \forall \Delta \in \frac{1}{\alpha} \bar{\mathcal{B}}_\Delta \quad (2.42)$$

Démonstration du théorème 2.6 : Soit $C = B\hat{C}B^* - \epsilon EE^*$. On a l'égalité suivante :

$$C^*(I_n - \Delta M) = \begin{bmatrix} B & E \end{bmatrix} \begin{bmatrix} \hat{C}^*(I_q - B^* \Delta A) & 0 \\ \epsilon E^* \Delta A & -\epsilon I \end{bmatrix} \begin{bmatrix} B^* \\ E^* \end{bmatrix} \quad (2.43)$$

Ainsi :

$$\text{He } C^*(I_n - \Delta M) = \begin{bmatrix} B & E \end{bmatrix} \underbrace{\begin{bmatrix} \text{He } \hat{C}^*(I_q - B^* \Delta A) & \frac{\epsilon}{2} A^* \Delta E \\ \frac{\epsilon}{2} E^* \Delta A & -\epsilon I \end{bmatrix}}_R \begin{bmatrix} B^* \\ E^* \end{bmatrix} \quad (2.44)$$

Après application du complément de Schur, on obtient :

$$R < 0 \Leftrightarrow \begin{cases} -\epsilon I < 0 \\ \text{He } \hat{C}^*(I_q - B^* \Delta A) + \frac{\epsilon}{4} (A^* \Delta E)(A^* \Delta E)^* < 0 \end{cases} \quad (2.45)$$

La première condition de la partie droite de (2.45) est redondante, et la seconde est de toute évidence satisfaite pour un $\epsilon > 0$ assez petit, entraînant $R < 0$. Et puisque par hypothèse $\begin{bmatrix} B & E \end{bmatrix}$ est de rang plein, il peut alors être affirmé que $\text{He } C^*(I_n - \Delta M) < 0$. ■

Remarque 2.11

Pour appliquer le théorème 2.6, il faut veiller à choisir ϵ assez petit pour ne pas violer la condition (2.45) en dégradant trop le multiplieur, et avoir E tel que $\begin{bmatrix} B & E \end{bmatrix}$ soit de rang plein. Numériquement, il est en fait très facile de satisfaire cette dernière condition sur E en choisissant une matrice pleine générée aléatoirement, et en testant la condition sur le rang.

L'algorithme 4 est maintenant proposé afin de calculer une borne supérieure de μ_Δ , garantie sur tout l'intervalle de fréquences \mathbb{R}^+ , utilisant à la fois les théorèmes précédents ainsi que les μ -sensibilités.

Algorithme 4: Calcul d'une borne supérieure de μ_Δ par l'approche par multi-
plieurs

Entrées : $M(s)$, Δ , r_1

Sorties : ν_Δ^p

- 1 Initialiser $\mathcal{I} = \{\mathbb{R}^+\}$ la liste des intervalles fréquentiels sur laquelle la borne supérieure doit être évaluée ;
 - 2 Initialiser $\nu_{\max} = 0$;
 - tant que** $\mathcal{I} \neq \emptyset$
 - 3 Choisir un intervalle $I \in \mathcal{I}$ et une fréquence $\omega \in I$;
 - 4 Résoudre le problème (2.32) afin de calculer $\nu_\Delta^p(M(j\omega))$ et un multiplieur $\hat{C} \in \mathbb{C}^{q \times q}$;
 - si** $q < n$ **alors**
 - 5 Utiliser le théorème 2.6 afin de calculer un multiplieur de rang plein $C \in \mathbb{C}^{n \times n}$;
 - sinon**
 - 6 Poser $C = \hat{C}$;
 - 7 Définir $\nu_{\max} = \max(\nu_{\max}, \nu_\Delta^p(M(j\omega)))$;
 - 8 Appliquer le théorème 2.5 pour chaque $\Delta_j \in \frac{1}{\nu_{\max}} \bar{\mathcal{B}}_\Delta$ afin d'évaluer les intervalles fréquentiels \mathcal{I}_{v_j} pour lesquels la borne supérieure ν_{\max} et le multiplieur C sont valides ;
 - 9 Calculer $\mathcal{I}_v = \bigcap_j \mathcal{I}_{v_j}$;
 - 10 Mettre à jour $\mathcal{I} \leftarrow \mathcal{I} \setminus \mathcal{I}_v$;
 - 11 $\nu_\Delta^p = \nu_{\max}$ est une borne supérieure de μ_Δ qui vérifie $\nu_\Delta \leq \nu_\Delta^p \leq \bar{\mu}_\Delta^{\text{LMI}}$.
-

2.3.4 Résultats

Cette sous-section présente les résultats obtenus à l'aide de l'algorithme 4, sur une liste de benchmarks pertinents extraits du tableau 1.1. Ces derniers ont point commun que l'approche par matrices (D,G) ne donne pas une borne supérieure de μ_Δ satisfaisante. Les résultats sont présentés sur le tableau 2.5, qui contient :

- le numéro du benchmark (ses caractéristiques sont regroupées dans le tableau 1.1),
- la meilleure borne inférieure $\underline{\mu}_\Delta$ de μ_Δ sur \mathbb{R}^+ parmi tous les algorithmes existants,
- la borne supérieure $\bar{\mu}_\Delta^{\text{LMI}}$ obtenue par l'approche par matrices (D,G) sur \mathbb{R}^+ avec le temps de calcul correspondant, et l'écart η entre $\underline{\mu}_\Delta$ et $\bar{\mu}_\Delta^{\text{LMI}}$,
- la borne supérieure ν_Δ^p obtenue par l'approche par multiplieurs avec l'algorithme 4, avec le nombre de blocs incertains r_1 optimisés, le temps de calcul,

le nombre d'itérations nécessaire pour valider la borne sur \mathbb{R}^+ , et l'écart entre $\underline{\mu}_\Delta$ et ν_Δ^p .

Benchmark	$\underline{\mu}_\Delta$	Matrices (D,G)			Multiplieurs				
		$\bar{\mu}_\Delta^{\text{LMI}}$	Temps (s)	η	r_1	ν_Δ^p	Temps (s)	Itérations	η
1	1.711	2.413	1	41.03%	2	1.732	42	27	1.23%
2	1.620	2.067	6	27.59%	2	1.850	120	50	14.20%
					4	1.622	477	89	0.09%
5	0.184	0.229	11	24.46%	3	0.222	100	4	20.38%
					5	0.208	133	4	12.88%
					9	0.194	1190	6	5.34%

TABLEAU 2.5 – Calcul d'une borne supérieure de μ_Δ en utilisant l'approche par multiplieurs (algorithme 4), et comparaison avec l'approche par matrices (D,G)

Les résultats montrent bien une forte réduction du conservatisme avec un temps de calcul qui reste très raisonnable. Pour le benchmark 5 il est important de souligner que, bien que la taille du modèle soit grande, l'utilisation de la dégénérescence du rang permet de fortement diminuer la complexité numérique.

Il est également intéressant d'insister sur le faible nombre d'itérations nécessaires afin de valider ν_Δ^p sur la plage entière de fréquences, sans risque de manquer une fréquence critique. S'il avait fallu appliquer la méthode standard de [Fu et Barabanov 1997] avec un maillage fréquentiel, au moins 100 fréquences auraient dû être utilisées pour obtenir des résultats relativement précis. Ceci aurait impliqué un temps de calcul bien supérieur à ce qui est mentionné dans le tableau 2.5. Cette affirmation est illustrée sur le benchmark 1 par le tableau 2.6, où sont utilisés un maillage fin (100 fréquences) et un maillage grossier (30 fréquences). Dans le premier cas, le temps de calcul est clairement supérieur par rapport au tableau 2.5. Dans le second cas, le temps de calcul est plus raisonnable mais la borne supérieure de μ_Δ obtenue n'est pas valide puisque inférieure à la meilleure borne inférieure de μ_Δ connue.

Benchmark	Maillage	ν_Δ^p	Temps (s)
1	Fin (100 fréquences)	1.7255	115
	Grossier (30 fréquences)	1.5715	38

TABLEAU 2.6 – Calcul d'une borne supérieure de μ_Δ par l'approche par multiplieurs sur un maillage fréquentiel.

CONCLUSION DU CHAPITRE

Pour beaucoup d'applications réalistes, les techniques les plus communément utilisées afin de calculer une borne supérieure de μ_Δ sont soit conservatives, soit lourdes en termes de temps de calcul. Dans ce chapitre, trois nouveaux algorithmes ont été proposés afin de pallier ces problèmes. Ils sont à utiliser dans les cas suivants :

- la résolution partielle des LMIs pour l'approche par matrices (D,G) de [Young et al. 1995] est conseillée si la valeur de la borne supérieure de μ_Δ change drastiquement selon qu'elle est calculée avec un solveur LMI, ou avec une approche de descente de gradient,
- la nouvelle technique de branch-&-bound peut être utilisée lorsque le conservatisme de la borne supérieure utilisant l'approche classique par matrices (D,G) est trop élevé,
- l'approche par multiplieurs avec validation sur l'ensemble des intervalles fréquents est également à employer lorsqu'un trop fort conservatisme de la borne supérieure de μ_Δ est révélé, mais sur des modèles qui restent de taille modérée car l'approche, bien que rendue plus rapide grâce aux μ -sensibilités et à l'utilisation d'un éventuel rang réduit du système, reste de complexité exponentielle.

Ces trois techniques utilisent toutes un outil commun permettant de détecter les incertitudes les plus significatives : les μ -sensibilités. Cependant rien n'a montré jusqu'ici que l'utilisation des μ -sensibilités permet effectivement une nette amélioration par rapport à un choix "aléatoire" des incertitudes. Ce point est clarifié dans le tableau 2.7.

Méthode	Grandeur comparée	Bench	Avec les μ -sensibilités	Choix aléatoires (50 tirages)		
				Min	Moyenne	Max
LMIs partielles (matrices (D,G)) Algorithme 2	$\frac{\bar{\mu}_\Delta^{\text{PLMI}} - \bar{\mu}_\Delta^{\text{LMI}}}{\bar{\mu}_\Delta^{\text{LMI}}}$	7	0.01%	3.00%	8.80%	15.06%
		8	0.57%	5.54%	10.19%	17.50%
		11	97.34%	117.66%	183.94%	208.91%
Branch & bound Algorithme 3	Temps CPU ($\eta = 5\%$)	5	180s	374s	1654s	3689s
		6	481s	1751s	2439s	3571s
Multiplieurs Algorithme 4	$\frac{v_\Delta^p - \mu_\Delta}{\mu_\Delta}$	2	14.20%	20.55%	20.84%	20.93%
		5	5.34%	5.34%	11.49%	17.88%

TABLEAU 2.7 – Résultats montrant l'intérêt d'utiliser les μ -sensibilités

Il contient :

- la méthode appliquée : résolution partielle de LMIs (sous-section 2.2.1),

branch-&-bound (sous-section 2.2.2), ou approche par multiplieurs (section 2.3),

- la grandeur pertinente à comparer suivant la méthode : l'écart entre $\bar{\mu}_{\Delta}^{\text{LMI}}$ et $\bar{\mu}_{\Delta}^{\text{pLMI}}$ pour la résolution partielle de LMIs, le temps de calcul pour le branch-&-bound avec un écart désiré entre les bornes de 5%, et l'écart entre $\underline{\mu}_{\Delta}$ et ν_{Δ}^p pour l'approche par multiplieurs,
- le benchmark considéré,
- les valeurs obtenues en utilisant les μ -sensibilités,
- les valeurs minimale, moyenne, et maximale lorsque 50 tirages aléatoires des incertitudes à considérer sont effectués, afin de réaliser une étude statistique lorsque les μ -sensibilités ne sont pas utilisées.

Il est intéressant de remarquer que sur tous les exemples considérés, l'utilisation des μ -sensibilités permet d'obtenir les meilleurs résultats. Lorsque des choix aléatoires des incertitudes sont réalisés, les valeurs minimales ne sont généralement pas aussi bonnes. Ceci permet donc de parfaitement illustrer l'intérêt des μ -sensibilités.

Deuxième partie

Synthèse de lois de commande non-linéaires robustes

INTRODUCTION À L'INVERSION DYNAMIQUE NON-LINÉAIRE

SOMMAIRE

3.1	CAS DES SYSTÈMES CARRÉS AFFINES EN LA COMMANDE	71
3.1.1	Principe général	72
3.1.2	Limitations	73
3.2	EXTENSION : TECHNIQUE DE LINÉARISATION ENTRÉE-SORTIE	74
3.2.1	Cas des systèmes non-linéaires SISO	74
3.2.2	Systèmes avec échelles de temps multiples	78
3.2.3	Systèmes non-linéaires MIMO	80
3.3	TECHNIQUE DE ROBUSTIFICATION : SYNTHÈSE \mathcal{H}_∞	83
3.3.1	La problème \mathcal{H}_∞ standard	83
3.3.2	Quelques notes sur les fonctions de pondération	85
3.3.3	La synthèse \mathcal{H}_∞ structurée	87
	CONCLUSION	89

Résumé

Ce chapitre propose, sous forme d'état de l'art ciblé en fonction des besoins de cette thèse, une introduction aux techniques d'inversion dynamique. Cette méthode de commande des systèmes non-linéaires présente de nombreux avantages : possibilité de compenser les principales non-linéarités, de découpler une large classe de systèmes, et d'adapter la loi automatiquement au point de fonctionnement. Son efficacité dépend cependant beaucoup de la boucle externe, dont la synthèse utilise généralement des méthodes issues de la commande linéaire robuste. Il est proposé à ce propos de présenter notamment un bref rappel des techniques de synthèse \mathcal{H}_∞ , et notamment sa version structurée. Tout ceci servira de base pour les améliorations proposées au chapitre 4.

Les techniques de linéarisation entrée-sortie, introduites dans les années 1980 par [Isidori et al. 1981] et reprises dans de nombreux ouvrages et publications [Isidori 1985; Fliess 1985; Slotine et Li 1991; Vidyasagar 1993], permettent de linéariser et de découpler une large classe de systèmes non-linéaires. Elles ont été mises en œuvre sur des applications variées parmi lesquelles la commande des systèmes aéronautiques et spatiaux occupe une place de choix [Snell et al. 1992; Adams et Banda 1993; Reiner et al. 1995; 1996; Schumacher et Khargonekar 1998; Ito et al. 2002; Hernandez Lopezomoza 2012; Kara Mohamed et Lanzon 2013; 2012; Bianic et al. 2014]. L'inversion dynamique repose sur une structure à double boucles : une boucle interne de linéarisation visant à éliminer autant que possible les non-linéarités du système, et une boucle externe qui permet d'imposer les dynamiques du système découplé résultant.

Ce chapitre vise à présenter un bref état de l'art sur les techniques d'inversion dynamique, synthétisant la plupart des références mentionnées ci-avant. La section 3.1 introduit la technique sur un cas simple : la classe des systèmes non-linéaires carrés affines en la commande. Cette version simplifiée permet d'illustrer clairement la structure à double boucles imbriquées, où la boucle interne permet notamment d'obtenir une forme linéaire standard appelée forme de Brunovsky (constituée d'une chaîne d'intégrateurs). Les limitations de la méthode, qu'on cherchera à corriger dans le chapitre 4, seront également mentionnées. La section 3.2 généralise la méthode à une plus large classe de systèmes, en présentant succinctement les techniques de linéarisation entrée-sortie. Une attention particulière sera portée sur le fait qu'il est parfois difficile d'inverser complètement le système, laissant des dynamiques non-commandées dont il faudra veiller à vérifier la stabilité (dynamique des zéros). Enfin dans la section 3.3 on s'attardera sur une technique de commande robuste linéaire : la synthèse \mathcal{H}_∞ , qui permet d'améliorer la robustesse de la boucle externe. Sa version structurée sera particulièrement intéressante, amenant des possibilités d'imposer l'ordre du correcteur, sa structure, ou encore de l'inclure dans un schéma de synthèse multi-canaux et multi-modèles.

3.1 CAS DES SYSTÈMES CARRÉS AFFINES EN LA COMMANDE

Afin d'illustrer brièvement de façon simple les principes de l'inversion dynamique non-linéaire, il est proposé ici de se focaliser sur une classe de systèmes particulière : les systèmes carrés, c'est-à-dire possédant autant d'entrées que de variables d'état, affines en la commande. Ils sont représentés par l'équation suivante :

$$\dot{x} = f(x) + G(x)u \quad (3.1)$$

où :

- $x \in \mathbb{R}^n$ est le vecteur d'états,
- $u \in \mathbb{R}^n$ est la commande,
- $f(x) \in \mathbb{R}^n$ et $G(x) \in \mathbb{R}^{n \times n}$ sont deux fonctions non-linéaires.

Afin de pouvoir appliquer les techniques d'inversion dynamique, l'hypothèse suivante doit être vérifiée.

Hypothèse 3.1

La fonction matricielle G est supposée non-singulière, signifiant que :

$$\det(G(x)) \neq 0, \forall x \in \mathcal{D} \subset \mathbb{R}^n \quad (3.2)$$

où \mathcal{D} représente l'ensemble des états admissibles du système.

3.1.1 Principe général

Le principe de l'inversion dynamique non-linéaire (qui sera désormais nommée sous l'abréviation NDI, pour *Nonlinear Dynamic Inversion* en anglais) consiste à utiliser la commande $u(x)$ afin de découpler et linéariser le système non-linéaire (3.1). Pour ce faire, il suffit d'utiliser explicitement les fonctions $f(x)$ et $G(x)$ afin d'éliminer les dynamiques non-linéaires, et ensuite de forcer la dynamique du système résultant. La loi est donc constituée de deux boucles, une boucle interne linéarisante, et une boucle externe stabilisante et permettant de répondre à certaines spécifications.

Boucle interne linéarisante

La boucle interne est définie par la loi de commande suivante :

$$u(x) = G^{-1}(x) (-f(x) + v) \quad (3.3)$$

en supposant bien entendu que tout l'état est accessible à la mesure.

Ainsi, en utilisant la commande (3.3) sur le système non-linéaire carré (3.1), on obtient le système résultant représenté par l'équation suivante :

$$\dot{x} = v \quad (3.4)$$

Ainsi, le système précédent est linéaire et est constitué d'une série de n intégrateurs.

Boucle externe

Une fois que les principales non-linéarités du système initial ont été compensées par la loi interne linéarisante, la commande v apparue à l'équation (3.4) peut être utilisée afin d'imposer une dynamique désirée au système bouclé. La structure suivante est généralement employée :

$$v = K(s) \begin{bmatrix} x_c \\ x \end{bmatrix} \quad (3.5)$$

où x_c représente le vecteur de consignes à suivre, et $K(s)$ est le correcteur qui doit être dimensionné afin d'imposer une dynamique au système résultant ainsi qu'un suivi des consignes. La loi de commande complète est alors donnée par :

$$u(x) = G^{-1}(x) \left(-f(x) + K(s) \begin{bmatrix} x_c \\ x \end{bmatrix} \right) \quad (3.6)$$

Puisque le système (3.4) est linéaire, toute méthode de synthèse linéaire peut être employée afin de synthétiser le correcteur $K(s)$. On peut citer le placement de pôles, la commande par PI/PD/PID, optimale LQ, ou des techniques plus avancées comme la commande robuste par synthèse \mathcal{H}_∞ qui sera décrite à la section 3.3.

3.1.2 Limitations

La commande par inversion dynamique non-linéaire classique, telle qu'exposée dans la présente section, semble assez attrayante et simple à mettre en place, et permet de traiter efficacement les non-linéarités et le découplage sur un système non-linéaire. Cependant il existe plusieurs limitations qui peuvent s'avérer gênantes pour son utilisation dans des applications réalistes :

- le système (3.1) est loin d'être représentatif de tous les cas de figure et peut s'avérer extrêmement simplifié,
- aucune dynamique des actionneurs n'est prise en compte, tout comme les retards et dynamique des capteurs, qui peuvent entraîner des limitations en pratique, comme empêcher une linéarisation exacte du système initial,
- il n'est pas non plus évident de savoir si toutes les non-linéarités doivent être compensées, certaines pouvant en effet être bénéfiques pour la dynamique du système : dépenser de l'énergie de commande afin de les rejeter peut s'avérer peu recommandable,
- la NDI nécessite l'accès complet à l'état du système, ce qui peut s'avérer restrictif dans de nombreux cas,
- un des points les plus critiques concerne surtout la validité des fonction $f(x)$

et $G(x)$ qui ne sont que des modélisations. En pratique, diverses incertitudes affecteront ces deux termes, rendant d'une part impossible toute linéarisation exacte et d'autre part nécessaire la robustesse de la loi externe.

Ainsi un gros effort peut être demandé à la boucle externe afin de compenser ces limitations, qui doit alors s'avérer extrêmement robuste tout en assurant un bon suivi de consignes.

Dans la section suivante, on s'intéresse au cas un peu plus général où le système considéré n'est plus nécessairement carré, c'est-à-dire que le nombre d'entrées n'est plus égal au nombre d'états du système. Des notions sur la technique de linéarisation entrée-sortie sont à cet effet introduites afin d'obtenir autant que possible une boucle interne linéarisante.

3.2 EXTENSION : TECHNIQUE DE LINÉARISATION ENTRÉE-SORTIE

3.2.1 Cas des systèmes non-linéaires SISO

Dans cette sous-section, un rappel sur la théorie de la commande par linéarisation entrée-sortie est introduit, appliquée aux système mono-entrée / mono-sortie (*Single Input Single Output* en anglais, i.e. SISO).

On considère la classe des systèmes non-linéaires SISO affines en la commande décrits sous la forme générale suivante :

$$\begin{aligned}\dot{x} &= f(x) + g(x)u \\ z &= h(x)\end{aligned}\tag{3.7}$$

où $x \in \mathbb{R}^n$ est le vecteur d'état, $z \in \mathbb{R}$ est la sortie, $u \in \mathbb{R}$ est la commande, f et g sont des champs de vecteurs et h est une fonction à valeurs dans \mathbb{R} . Le but de la commande par linéarisation entrée-sortie est de déterminer un changement de coordonnées $\tilde{x} = \Phi(x)$ et un retour d'état statique transformant le système (3.7) en un système linéaire, découplé et commandable.

Définition 3.1 (Linéarisation entrée-sortie)

Le système non-linéaire (3.7) peut être transformé en un système linéaire par une commande par linéarisation entrée-sortie s'il existe un difféomorphisme $\tilde{x} = \Phi(x)$ et une commande :

$$u(x) = a(x) + b(x)v \quad (3.8)$$

où $a(x)$ et $b(x) \neq 0$ sont des fonctions scalaires réelles assurant l'existence d'une matrice $A \in \mathbb{R}^{n \times n}$ et de vecteurs $B \in \mathbb{R}^{n \times 1}$, $C \in \mathbb{R}^{1 \times n}$ tels que la boucle fermée s'écrive :

$$\begin{cases} \dot{\tilde{x}} = A\tilde{x} + Bv \\ z = C\tilde{x} \end{cases} \quad (3.9)$$

Il reste maintenant à déterminer une telle commande linéarisante. Le principe consiste à dériver successivement la sortie régulée z jusqu'à faire apparaître de façon explicite l'entrée u . On définit tout d'abord la dérivée de Lie, dénotée $L_f h(x)$, afin d'exprimer la dérivée d'une fonction scalaire $h(x)$ par rapport à un champ de vecteurs $f(x)$.

Définition 3.2 (Dérivée de Lie)

Soient un champ de vecteurs $f(x) \in \mathbb{R}^n$ et une fonction scalaire $h(x) \in \mathbb{R}$, avec $x \in \mathbb{R}^n$. On définit la dérivée de Lie de $h(x)$ par rapport à $f(x)$ par :

$$L_f h(x) = \frac{\partial h(x)}{\partial x} f(x) \quad (3.10)$$

$L_f h(x)$ correspond à la dérivée de h le long des trajectoires du système défini par $\dot{x} = f(x)$.

Le nombre minimum r de dérivations successives de z nécessaires à l'apparition de la commande u est appelé degré relatif.

Définition 3.3 (Degré relatif d'un système non-linéaire)

Le système non-linéaire (3.7) a un degré relatif r si :

- pour tout entier $i < r - 1$, $L_g L_f^i h(x) = 0$
- $L_g L_f^{r-1} h(x) \neq 0$

En supposant que le degré relatif du système (3.7) vaut r , les r dérivées successives de la sortie régulée z s'écrivent de la façon suivante :

$$\begin{cases} z &= h(x) \\ \dot{z} &= L_f h(x) \\ &\vdots \\ z^{(r-1)} &= L_f^{r-1} h(x) \\ z^{(r)} &= L_f^r h(x) + L_g L_f^{r-1} h(x)u \end{cases} \quad (3.11)$$

On observe bien l'apparition de la commande u lors de la r -ième dérivée de z . On peut donc introduire le changement de variable $\tilde{x} = \Phi(x)$ tel que :

$$\tilde{x} = \Phi(x) = \begin{bmatrix} \tilde{x}_1(x) \\ \tilde{x}_2(x) \\ \vdots \\ \tilde{x}_{r-1}(x) \\ \tilde{x}_r(x) \end{bmatrix} = \begin{bmatrix} z \\ \dot{z} \\ \vdots \\ z^{(r-2)} \\ z^{(r-1)} \end{bmatrix} = \begin{bmatrix} h(x) \\ L_f h(x) \\ \vdots \\ L_f^{r-2} h(x) \\ L_f^{r-1} h(x) \end{bmatrix} \quad (3.12)$$

conduisant le système non-linéaire (3.7) au nouveau système dont la dynamique est donnée par :

$$\begin{cases} \dot{\tilde{x}}_1 &= \tilde{x}_2 \\ \dot{\tilde{x}}_2 &= \tilde{x}_3 \\ &\vdots \\ \dot{\tilde{x}}_{r-1} &= \tilde{x}_r \\ \dot{\tilde{x}}_r &= L_f^r h(x) + L_g L_f^{r-1} h(x) u \\ \dot{\eta} &= \psi(\tilde{x}, \eta) \\ z &= \tilde{x}_1 \end{cases} \quad (3.13)$$

Les non-linéarités se trouvent à présent dans l'équation de $\dot{\tilde{x}}_r$, où la commande u est présente. Il est à noter également que le degré relatif r pouvant être inférieur à l'ordre du système n , il peut persister une dynamique interne, représentée par l'équation $\dot{\eta} = \psi(\tilde{x}, \eta)$ avec $\psi(\tilde{x}, \eta) \in \mathbb{R}^{n-r}$, qui ne dépend pas de la commande et qui ne peut par conséquent pas être commandée. Ce sujet est discuté plus en détails dans la remarque 3.1.

Puisque $L_g L_f^{r-1} h(x) \neq 0$, on peut utiliser la commande u afin de compenser les non-linéarités de (3.13) avec :

$$u(x) = \frac{1}{L_g L_f^{r-1} h(x)} \left(-L_f^r h(x) + v \right) \quad (3.14)$$

En identifiant dans l'équation (3.8), la structure de la loi peut être récupérée avec :

$$a(x) = \frac{-1}{L_g L_f^{r-1} h(x)} L_f^r h(x), \quad b(x) = \frac{1}{L_g L_f^{r-1} h(x)} \quad (3.15)$$

Ceci permet donc d'établir une relation linéaire entre la nouvelle entrée v et la sortie

régulée z . Le système résultant est le suivant :

$$\left\{ \begin{array}{lcl} \dot{\tilde{x}}_1 & = & \tilde{x}_2 \\ \dot{\tilde{x}}_2 & = & \tilde{x}_3 \\ & \vdots & \\ \dot{\tilde{x}}_{r-1} & = & \tilde{x}_r \\ \dot{\tilde{x}}_r & = & v \\ \dot{\eta} & = & \psi(\tilde{x}, \eta) \\ z & = & \tilde{x}_1 \end{array} \right. \quad (3.16)$$

qui est constitué d'une chaîne de r intégrateurs. La forme du système représenté par (3.16) est appelée **forme canonique de Brunovsky**. La commande v peut alors être déterminée en utilisant des techniques de la théorie de la commande linéaire, afin de stabiliser et d'imposer une dynamique désirée au système linéaire résultant.

Remarque 3.1

Puisque le système obtenu est d'ordre r , il est important de le comparer à l'ordre n du système d'origine, ce qui permet d'en tirer 3 cas :

1. $r = n$: il est possible de reconstituer totalement le système non-linéaire (3.7) à partir de la seule sortie z , sans ignorer d'éventuelles dynamiques internes. C'est le cas le plus adapté à la commande par linéarisation entrée-sortie. Le terme dans (3.16), $\dot{\eta} = \psi(\tilde{x}, \eta)$, n'est donc pas présent dans ce cas.
2. $r < n$: le système non-linéaire d'origine contient des dynamiques internes représentées par l'équation $\dot{\eta} = \psi(\tilde{x}, \eta)$ dans (3.16), qui ne sont pas observables. En particulier, la dynamique correspondant à $\dot{\eta} = \psi(0, \eta)$ correspond à la dynamique des zéros du système. Elle représente la dynamique interne du système (3.7) lorsque la commande u contraint la sortie $z = h(x)$ à être nulle. Deux cas peuvent alors se présenter :
 - Si la dynamique des zéros est asymptotiquement stable, la loi de commande décrite par l'équation (3.14) linéarise et découple le système (3.7) entre l'entrée v et la sortie régulée z .
 - Si la dynamique des zéros est instable, le système (3.7) est dit à non-minimum de phase et la dynamique inverse est instable. Il n'est alors pas possible d'utiliser une commande par linéarisation entrée-sortie garantissant la stabilité interne du système résultant. Un changement de la sortie régulée peut éventuellement permettre d'observer toutes les dynamiques internes et peut être une solution à adopter dans ce cas.
3. $r > n$: il est impossible de faire apparaître la commande à partir des sorties, et une commande par linéarisation entrée-sortie ne peut être déterminée.

Exemple 3.1

Afin d'illustrer la technique de linéarisation entrée-sortie pour un système SISO, on introduit le modèle longitudinal d'avion de chasse à 2 états suivant :

$$\begin{cases} \dot{\alpha} &= w_{\alpha}(\alpha) + q \\ \dot{q} &= w_q(\alpha, q) + m_{\delta}\delta_e \end{cases} \quad (3.17)$$

où α représente l'angle d'incidence, q la vitesse de tangage, δ_e la commande de déflection des gouvernes de profondeur, $m_{\delta} \neq 0$ l'efficacité de la gouverne, et w_{α} et w_q des fonctions non-linéaires de α et q .

On cherche à commander l'angle d'incidence α , on a donc la sortie régulée $z = \alpha$. On obtient par dérivations successives :

$$\begin{aligned} z &= \alpha \\ \dot{z} &= w_{\alpha}(\alpha) + q \\ \ddot{z} &= \dot{w}_{\alpha}(\alpha) + w_q(\alpha, q) + m_{\delta}\delta_e \end{aligned} \quad (3.18)$$

Puisque la commande δ_e apparaît dans l'expression de \ddot{z} , le degré relatif est de 2 et est égal à l'ordre du système. Il est donc possible de déterminer une commande par linéarisation entrée-sortie. La commande linéarisante est donnée par :

$$\delta_e = -\frac{1}{m_{\delta}} (\dot{w}_{\alpha}(\alpha) + w_q(\alpha, q) + v) \quad (3.19)$$

En utilisant la matrice de changement de variable :

$$\Phi(\alpha, q) = \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} = \begin{bmatrix} \alpha \\ w_{\alpha}(\alpha) + q \end{bmatrix} \quad (3.20)$$

on obtient le système bouclé transformé suivant :

$$\begin{bmatrix} \dot{\tilde{x}}_1 \\ \dot{\tilde{x}}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v \quad (3.21)$$

qui correspond à un système linéaire avec deux intégrateurs et une nouvelle entrée v . Il suffit alors d'utiliser une méthode de synthèse linéaire afin de dimensionner v (placement de pôles, correcteur PID, ...).

3.2.2 Systèmes avec échelles de temps multiples

Sur certains systèmes SISO, il est possible de décomposer leur dynamique en deux voire plusieurs sous-systèmes qui peuvent être alors traités en cascade par inversion dynamique. Il faut dans ce cas que les échelles de temps de ces sous-

systèmes soient suffisamment séparées, afin de les commander successivement. On représente cette classe de systèmes par le système dynamique suivant :

$$\begin{cases} \dot{x}_1 = f_1(x_1) + g_1(x_1)x_2 \\ \dot{x}_2 = f_2(x_1, x_2) + g_2(x_1, x_2)x_3 \\ \vdots \\ \dot{x}_{k-1} = f_{k-1}(x_1, \dots, x_{k-1}) + g_{k-1}(x_1, \dots, x_{k-1})x_k \\ \dot{x}_k = f_k(x) + g_k(x)u \end{cases} \quad (3.22)$$

où la dynamique de x_1 est considérée la plus lente, alors que celle de x_k est la plus rapide, et $x = [x_1 \dots x_k]^T$ est le vecteur d'états. Étant donné que la commande u apparaît au niveau de la dynamique de l'état x_k , il est possible d'utiliser une commande par NDI :

$$u = g_k^{-1}(x) \left(-f_k(x) + K_k(s) \begin{bmatrix} x_{k_c} \\ x_k \end{bmatrix} \right) \quad (3.23)$$

Une dynamique peut alors être imposée sur l'état le plus rapide, qui influence de façon cascadée les états les plus lents. Le correcteur $K_k(s)$ assurant la convergence de x_k vers la consigne x_{k_c} , on obtient donc $x_k \approx x_{k_c}$ au bout d'un temps suffisant. L'état précédent x_{k-1} a donc ensuite comme dynamique :

$$\dot{x}_{k-1} = f_{k-1}(x_1, \dots, x_{k-1}) + g_{k-1}(x_1, \dots, x_{k-1})x_{k_c} \quad (3.24)$$

On peut alors voir x_{k_c} comme la nouvelle commande pour contrôler la dynamique de l'état x_{k-1} , afin d'amener l'état x_{k-1} à l'état commandé $x_{(k-1)_c}$ par inversion dynamique non-linéaire :

$$x_{k_c} = g_{k-1}^{-1} \left(-f_{k-1}(x_1, \dots, x_{k-1}) + K_{k-1}(s) \begin{bmatrix} x_{(k-1)_c} \\ x_{k-1} \end{bmatrix} \right) \quad (3.25)$$

Et on procède ainsi de suite jusqu'à l'état x_1 qui peut finalement être commandé. L'avantage de cette technique est sa relative simplicité, et le fait qu'un bon nombre de systèmes puisse être décomposé en dynamiques lente et rapide. Il faut cependant que la commande affecte directement la dynamique rapide, ce qui n'est pas du tout acquis dans beaucoup d'applications. De plus il est nécessaire également que l'état le plus lent soit celui qu'il est souhaitable de commander. L'exemple suivant, reprenant l'exemple 3.1, propose d'appliquer cette stratégie.

Exemple 3.2

On reprend le modèle longitudinal d'avion de chasse à 2 états illustré lors de l'exemple 3.1. La commande de braquage de gouvernes δ_e affecte directement la dynamique de la vitesse de tangage q , qui elle même affecte la dynamique de l'angle d'attaque α . De plus, la dynamique de vitesse de rotation étant plus rapide que la dynamique en position angulaire, on est alors bien situé dans le cadre de la séparation d'échelles de temps. On peut ainsi choisir dans un premier temps la commande suivante, afin d'imposer une dynamique du premier ordre de constante τ_q sur la vitesse de tangage q :

$$\delta_e = m_\delta^{-1} \left(-w_q(\alpha, q) + \frac{1}{\tau_q} (q_c - q) \right) \quad (3.26)$$

On a alors $\dot{q} = \frac{1}{\tau_q} (q_c - q)$. La dynamique étant suffisamment rapide, on va supposer que q a convergé vers la consigne q_c , menant alors à l'équation de la dynamique plus lente en α suivante :

$$\dot{\alpha} = w_\alpha(\alpha) + q_c \quad (3.27)$$

q_c peut alors être utilisé afin de commander l'angle d'attaque, par :

$$q_c = -w_\alpha(\alpha) + k_i \int (\alpha_c - \alpha) + k_p (\alpha_c - \alpha) \quad (3.28)$$

où α_c est la consigne, et k_i et k_p sont des gains correspondants à une commande PI, afin d'imposer une dynamique sur l'angle d'attaque de type second ordre, avec comme amortissement $\xi = \frac{k_p}{2\sqrt{k_i}}$, et comme fréquence propre $\omega = \sqrt{k_i}$.

La loi de commande permettant de commander l'angle d'attaque est donc la suivante :

$$\delta_e = m_\delta^{-1} \left(K(s) \begin{bmatrix} \alpha & q \end{bmatrix}^T + H(s) \begin{bmatrix} \alpha_c & w_\alpha & w_q \end{bmatrix}^T \right) \quad (3.29)$$

avec :

$$\begin{cases} K(s) &= \frac{1}{\tau_q} \left[2\xi\omega + \frac{\omega^2}{s} \quad 1 \right] \\ H(s) &= \begin{bmatrix} \frac{-\omega^2}{\tau_q s} & \frac{1}{\tau_q} & -1 \end{bmatrix} \end{cases} \quad (3.30)$$

La structure de la loi de commande précédente est intéressante. Le correcteur $K(s)$ amène des degrés de liberté supplémentaires en possédant comme entrées l'état complet du système. Quant au correcteur $H(s)$, il assure le suivi de consigne et également une rejection de perturbations mesurées, constitué des termes non-linéaires w_α et w_q . Le principe de cette stratégie de commande sera notamment exploitée au chapitre 4.

3.2.3 Systèmes non-linéaires MIMO

On s'intéresse dans cette section à la commande par linéarisation entrée-sortie dans le cas multi-entrées / multi-sorties (*Multiple Inputs Multiple Outputs* en anglais,

i.e. MIMO). Le système non-linéaire considéré est alors le suivant :

$$\begin{aligned}\dot{x} &= f(x) + \sum_{i=1}^p g_i(x)u_i \\ z &= h(x) = \begin{bmatrix} h_1(x) & \dots & h_p(x) \end{bmatrix}^T\end{aligned}\quad (3.31)$$

où $x \in \mathbb{R}^n$ est le vecteur d'état, $z \in \mathbb{R}^p$ est la sortie, $u_i \in \mathbb{R}$ sont les commandes du système avec $i = 1, \dots, p$, f et g_i sont des champs de vecteurs et les h_j sont des fonctions à valeurs dans \mathbb{R} , avec $j = 1, \dots, p$. Par soucis de clarté, la dynamique des zéros est ici négligée afin de ne pas surcharger les équations qui vont suivre. Cependant, l'analyse sur celle-ci est la même que celle sur les systèmes non-linéaires SISO.

Afin de transformer le système (3.31) en un système découplé et totalement (ou partiellement dans le cas où la dynamique des zéros est présente) linéarisé, la même procédure que pour les systèmes non-linéaires SISO est appliquée, à chaque transfert entre les entrées u_i et les sorties z_j . Il suffit donc d'appliquer le changement de variable représenté précédemment par le difféomorphisme $\Phi(x)$ à chaque sortie j . En notant r_i le degré relatif associé, correspondant au nombre minimum de dérivations de la sortie z_j requis pour faire apparaître au moins une des entrées u_i à la r_j -ème dérivation, la nouvelle application est donc de la forme suivante :

$$\tilde{X} = \begin{bmatrix} \tilde{x}^1 \\ \vdots \\ \tilde{x}^p \end{bmatrix} = \begin{bmatrix} \tilde{x}_1^1(x) \\ \vdots \\ \tilde{x}_{r_1}^1(x) \\ \tilde{x}_1^p(x) \\ \vdots \\ \tilde{x}_{r_p}^p(x) \end{bmatrix} = \begin{bmatrix} z_1 \\ \vdots \\ z_1^{(r_1-1)} \\ \vdots \\ z_p \\ \vdots \\ z_p^{(r_p-1)} \end{bmatrix} = \begin{bmatrix} h_1(x) \\ \vdots \\ L_f^{r_1-1}h_1(x) \\ \vdots \\ h_p(x) \\ \vdots \\ L_f^{r_p-1}h_p(x) \end{bmatrix}\quad (3.32)$$

et conduit au système dont la dynamique est décrite par :

$$\begin{bmatrix} \dot{\tilde{x}}^1 \\ \vdots \\ \dot{\tilde{x}}^p \end{bmatrix} = \begin{bmatrix} \dot{\tilde{x}}_1^1 \\ \vdots \\ \dot{\tilde{x}}_{r_1}^1 \\ \vdots \\ \dot{\tilde{x}}_1^p \\ \vdots \\ \dot{\tilde{x}}_{r_p}^p \end{bmatrix} = \begin{bmatrix} \tilde{x}_2^1 \\ \vdots \\ L_f^{r_1} h_1(x) + \begin{bmatrix} L_{g_1} L_f^{r_1-1} h_1(x) & \dots & L_{g_p} L_f^{r_1-1} h_1(x) \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_p \end{bmatrix} \\ \hline \vdots \\ \tilde{x}_2^p \\ \vdots \\ L_f^{r_p} h_p(x) + \begin{bmatrix} L_{g_1} L_f^{r_p-1} h_p(x) & \dots & L_{g_p} L_f^{r_p-1} h_p(x) \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_p \end{bmatrix} \end{bmatrix} \quad (3.33)$$

Après avoir regroupé toutes les équations de (3.33) contenant la commande, on peut en déduire la loi de linéarisation entrée-sortie qui linéarise le système (3.31) :

$$u = \begin{bmatrix} u_1 \\ \vdots \\ u_p \end{bmatrix} = \underbrace{\begin{bmatrix} L_{g_1} L_f^{r_1-1} h_1(x) & \dots & L_{g_p} L_f^{r_1-1} h_1(x) \\ \vdots & \ddots & \vdots \\ L_{g_1} L_f^{r_p-1} h_p(x) & \dots & L_{g_p} L_f^{r_p-1} h_p(x) \end{bmatrix}^{-1}}_{G(x)^{-1}} \begin{bmatrix} -L_f^{r_1} h_1(x) + v_1 \\ \vdots \\ -L_f^{r_p} h_p(x) + v_p \end{bmatrix} \quad (3.34)$$

Le système linéarisé résultant de l'application de la loi de commande précédente est donc finalement donné par :

$$\begin{bmatrix} \dot{\tilde{x}}^1 \\ \vdots \\ \dot{\tilde{x}}^p \end{bmatrix} = \begin{bmatrix} \dot{\tilde{x}}_1^1(x) \\ \vdots \\ \dot{\tilde{x}}_{r_1}^1(x) \\ \vdots \\ \dot{\tilde{x}}_1^p(x) \\ \vdots \\ \dot{\tilde{x}}_{r_p}^p(x) \end{bmatrix} = \begin{bmatrix} \tilde{x}_2^1 \\ \vdots \\ v_1 \\ \vdots \\ \tilde{x}_2^p \\ \vdots \\ v_p \end{bmatrix} \quad (3.35)$$

Il est donc constitué d'une chaîne de $r = \sum_{i=1}^p r_i$ intégrateurs. Il a bien entendu été supposé que la matrice $G(x) \in \mathbb{R}^{p \times p}$ était inversible, ce qui est généralement le cas dans la plupart des applications. Certains problèmes peuvent cependant apparaître si les termes de la matrice $G(x)$ deviennent petits, pouvant entraîner un signal de commande trop élevé. De plus, la matrice $G(x)$ est également ici considérée carrée, du fait que l'on a autant d'entrées que de sorties, mais ce n'est généralement pas le

cas en pratique. Une solution à ce problème est d'utiliser de nouveau la séparation en échelle de temps introduite à la sous-section précédente.

Enfin il reste, tout comme le cas SISO, à choisir une commande $v = [v_1 \ \dots \ v_p]$ par des techniques linéaires, afin d'assurer les performances désirées en boucle fermée.

3.3 TECHNIQUE DE ROBUSTIFICATION : SYNTHÈSE \mathcal{H}_∞

On s'intéresse dans cette section à la technique de synthèse \mathcal{H}_∞ , introduite initialement par [Zames 1981] et développée ensuite par [Doyle et al. 1989]. Cette technique, inscrite dans le domaine de l'automatique linéaire, et plus particulièrement dans les méthodes de commande robuste (voir [Zhou et al. 1996; Zhou et Doyle 1998]), est traduit sous la forme un problème de rejection de perturbations externes tout en assurant la stabilisation interne du système commandé. Cette technique s'avère efficace et très flexible pour fixer des spécifications, via l'utilisation de fonctions de pondérations fréquentielles sur certains transferts notamment, qui traduisent des contraintes temporelles. Elle sera ainsi très fortement considérée dans ces travaux de thèse lorsqu'il s'agira de dimensionner la loi de commande externe après la phase de linéarisation par inversion dynamique. On s'attardera notamment sur la présentation de la version structurée qui permettra à la fois de considérer plusieurs canaux et objectifs, et également plusieurs modèles de synthèse afin d'améliorer la robustesse de la loi synthétisée sans excès de conservatisme.

Cette section ne se veut pas exhaustive mais propose quelques bases pour la compréhension de la technique mise en œuvre au chapitre 4. Le lecteur intéressé trouvera dans [Zhou et Doyle 1998; Alazard et al. 1999; Apkarian et Noll 2006] des compléments.

3.3.1 La problème \mathcal{H}_∞ standard

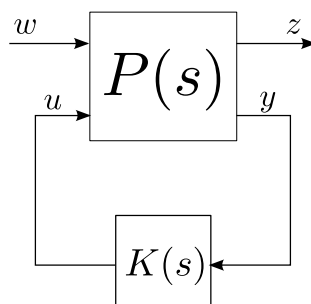


FIGURE 3.1 – Problème \mathcal{H}_∞ standard

On considère le problème \mathcal{H}_∞ standard représenté sur la figure 3.1, où :

- $P(s)$ représente le système linéaire augmenté, comprenant notamment le système nominal, des pondérations fréquentielles, et d'éventuels modèles de référence,
- w est le vecteur des entrées exogènes, qui peut contenir les consignes et les perturbations externes,
- z est le vecteur des sorties exogènes pondérées, qui peut contenir notamment l'erreur entre la sortie régulée et la sortie d'un modèle de référence.

Il est à noter que l'interconnexion de la figure 3.1 correspond à la LFT inférieure $\mathcal{F}_l(P(s), K(s))$ telle que définie dans le chapitre 1.

Le problème de synthèse \mathcal{H}_∞ consiste donc à minimiser le transfert entre w et z , noté $\mathcal{T}_{w \rightarrow z}(s)$, au sens de la norme \mathcal{H}_∞ , définie comme suit :

Définition 3.4 (Norme \mathcal{H}_∞)

Soit $G(s)$ la fonction de transfert d'un système stable, ayant comme vecteur d'entrée w et comme vecteur de sortie z . Sa norme \mathcal{H}_∞ , notée $\|G(s)\|_\infty$ est donnée par :

$$\|G(s)\|_\infty = \sup_{\omega \in \mathbb{R}} \bar{\sigma}(G(j\omega)) = \sup_{w \in \mathcal{L}_2} \frac{\|z\|_2}{\|w\|_2} \quad (3.36)$$

où \mathcal{L}_2 désigne l'espace de Hilbert pour les fonctions de carré intégrable au sens de l'intégrale de Lebesgue.

Ainsi la minimisation de ce transfert implique un impact limité des perturbations w considérées bornées en énergie sur les sorties exogènes z . Déterminer le correcteur :

$$\hat{K}(s) = \underset{K(s)}{\operatorname{argmin}} \|\mathcal{F}_l(P(s), K(s))\|_\infty \quad (3.37)$$

permet donc de réaliser de la rejection de perturbations tout en assurant par exemple un suivi de modèle de référence selon la nature des signaux exogènes w et z .

La théorie du faible gain et ses variantes donnent aussi un autre sens à la synthèse \mathcal{H}_∞ , justifiant son appartenance à la classe des techniques de commande robuste. Supposons en effet que les signaux w et z soient reliés via l'opérateur incertain Δ , de sorte que $w = \Delta(z)$, et que le correcteur $K(s)$ garantisse la propriété :

$$\|\mathcal{F}_l(P(s), K(s))\|_\infty < \gamma \quad (3.38)$$

Alors, la boucle fermée non-linéaire incertaine reste stable pour tout opérateur Δ γ^{-1} -Lipschitzien tel que :

$$\|\Delta(z)\|_\infty < \frac{1}{\gamma} \|z\|_\infty \quad (3.39)$$

Autrement dit, plus la norme \mathcal{H}_∞ d'un transfert est faible, pour la tolérance aux

incertitudes vues par ce transfert est grande.

La combinaison des deux aspects que nous venons de citer offre des possibilités multiples de synthèse de lois de commande performantes et robustes, qui ont fait le succès de cette méthodologie depuis la fin des années 80.

Sur le plan numérique pourtant, cette approche souffre jusqu'en 2006 d'un point faible majeur qui rend son utilisation parfois difficile. Ce point faible concerne l'ordre souvent élevé (celui de la forme standard pondérée $P(s)$) et le manque de lisibilité des correcteurs obtenus.

En effet, le problème d'optimisation (3.37) n'est convexe (et donc admet une solution globale que l'on sait calculer par résolution d'équations de Ricatti couplées (DGKE, [Doyle et al. 1989]) ou par approche LMI [Gahinet et Apkarian 1994]) que si le correcteur $K(s)$ est non-structuré avec un nombre d'états identique à celui du modèle de synthèse. Dans certaines applications il se peut aussi que le correcteur optimal soit instable, ce qui est critique en présence de saturations voire de panne d'un ou de plusieurs actionneurs.

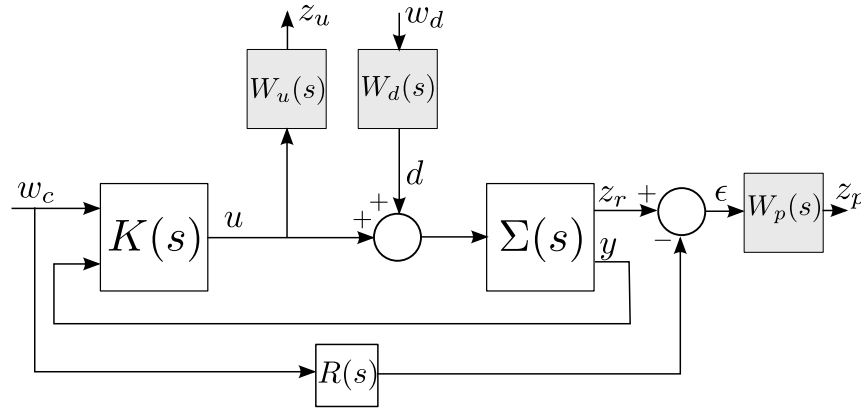
Depuis 2006 la synthèse \mathcal{H}_∞ connaît cependant un regain d'intérêt dans l'industrie avec la mise au point d'algorithmes de résolution utilisant les techniques d'optimisation non-lisses. Ces algorithmes permettent de déterminer très efficacement des optima locaux de très bonne qualité au problème d'optimisation \mathcal{H}_∞ structuré, d'ordre fixé. Nous y reviendrons dans la sous-section 3.3.3.

3.3.2 Quelques notes sur les fonctions de pondération

Jusqu'ici, rien n'a été vu par rapport à ce qui permet de fixer des spécifications fréquentielles afin de répondre à des contraintes temporelles, pour obtenir un système commandé satisfaisant à certaines exigences. Le système augmenté $P(s)$ comprend pour ce faire des fonctions de pondérations fréquentielles, qu'il est essentiel de convenablement dimensionner. Afin de donner quelques notions sur le choix de telles fonctions de pondération, on se place dans le cas classique d'un suivi de modèle de référence, représenté sur la figure 3.2 où :

- $\Sigma(s)$ est la fonction de transfert du système à commander, ayant pour sorties régulées z_r , pour mesures accessibles à la loi de commande y , et pour entrée $u + d$, u étant la commande et d un vecteur de perturbations,
- $R(s)$ est le modèle de référence, qui a pour entrées les consignes w_c ,
- $W_u(s)$, $W_d(s)$ et $W_p(s)$ sont les fonctions de pondération fréquentielles à dimensionner afin de répondre à des exigences sur le système bouclé.

Pour se ramener au schéma du problème \mathcal{H}_∞ standard figure 3.1, on a ici le vecteur d'entrées exogènes $w = \begin{bmatrix} w_c & w_d \end{bmatrix}^T$ et le vecteur de sorties exogènes pondérées

FIGURE 3.2 – Schéma de synthèse \mathcal{H}_∞

$z = \begin{bmatrix} z_p & z_u \end{bmatrix}^T$. Ainsi en minimisant la norme \mathcal{H}_∞ du transfert $\mathcal{T}_{w \rightarrow z}$, on minimise la norme \mathcal{H}_∞ de toutes les entrées vers toutes les sorties exogènes. Le système augmenté $P(s)$ sera de l'ordre du système $\Sigma(s)$, additionné avec les ordres de $R(s)$ et des 3 fonctions de pondération.

Le réglage des 3 fonctions de pondération nécessite généralement plusieurs essais afin de trouver le meilleur compromis. Il n'existe pas de règles absolues qui fonctionneront dans tous les cas. Cependant il est possible de donner quelques lignes directrices.

- La fonction $W_p(s)$, qui pondère l'erreur sur le suivi de modèle de référence $R(s)$, correspond à l'inverse du gabarit souhaité de la fonction de sensibilité. Elle doit être choisie comme un filtre passe-bas afin d'assurer une bonne précision à basses fréquences (typiquement la bande passante souhaitée du système).
- La fonction $W_u(s)$ permet de limiter l'activité de la commande. En choisissant un filtre passe-haut, les dynamiques hautes fréquences non-modélisées ne seront pas sollicitées, et la commande évitera notamment d'atteindre les saturations. Les choix concernant les valeurs du filtre dépendent essentiellement des capacités des actionneurs.
- La fonction $W_d(s)$ vise à réduire l'impact des perturbations externes sur le système et la commande. Le choix d'un gain statique dont la valeur est progressivement ajustée suffit amplement. Dans certains cas cependant, le profil fréquentiel de la perturbation est connu et peut être utilisé pour définir un filtre plus réaliste et moins conservatif.

Dans le cas MIMO, même si l'aspect directionnel peut intervenir, il est généralement préférable de choisir des matrices de pondération diagonales, où chaque terme correspondra à un transfert particulier.

3.3.3 La synthèse \mathcal{H}_∞ structurée

Les avantages de la synthèse \mathcal{H}_∞ sont clairs : elle tient compte dès la synthèse des perturbations externes, tout en permettant de spécifier des critères en performance (écart par rapport à un modèle de référence) et sur la commande (limitation de l'activité de celle ci à hautes fréquences).

Il existe cependant une limitation principale qui peut s'avérer gênante à son application pratique. L'ordre du correcteur obtenu est égal à l'ordre du système augmenté $P(s)$, qui contient celui du système auquel il faut ajouter ceux des pondérations. Un ordre de correcteur trop élevé peut poser des problèmes au moment de l'implantation sur le système réel, ou bien lorsqu'il faut interpoler plusieurs correcteurs notamment dans le cadre du séquençement de gains. Fixer l'ordre et la structure du correcteur rend le problème non-convexe. Diverses heuristiques ont été proposées pour calculer des solutions locales ne coïncidant pas toujours avec un optimum local.

Assez récemment, des avancées ont été réalisées dans le domaine de l'optimisation non-lisse appliquée à la stabilisation [Burke et al. 2006] et la synthèse \mathcal{H}_∞ [Apkarian et Noll 2006; 2007] structurée à ordre fixé. Ces avancées ont permis le développement de nouveaux algorithmes aujourd'hui disponibles sous Matlab et grâce auxquels des optima locaux de très bonne qualité sont souvent obtenus.

Plus récemment les algorithmes ont été étendus pour prendre en compte simultanément plusieurs canaux et plusieurs modèles [Gahinet et Apkarian 2011b;a].

Synthèse \mathcal{H}_∞ multi-canaux

Considérons le cas où le système $P(s)$ contient N entrées exogènes et N sorties exogènes pour simplifier les notations, de telle sorte que :

$$w = \begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix} \quad z = \begin{bmatrix} z_1 \\ \vdots \\ z_N \end{bmatrix} \quad (3.40)$$

Le problème \mathcal{H}_∞ standard cherche à minimiser γ tel que :

$$\left\| \begin{bmatrix} \mathcal{F}_l(P, K)_{11} & \dots & \mathcal{F}_l(P, K)_{1N} \\ \vdots & \ddots & \vdots \\ \mathcal{F}_l(P, K)_{N1} & \dots & \mathcal{F}_l(P, K)_{NN} \end{bmatrix} \right\|_\infty < \gamma \quad (3.41)$$

Ceci implique de considérer tous les termes, même les termes croisés. Or il peut s'avérer que tous les termes ne traduisent pas une spécification physique quelconque utile. Un des intérêts de la synthèse \mathcal{H}_∞ structurée est justement de pou-

voir fixer certains transferts nuls dans (3.41). On peut ainsi par exemple avoir à minimiser γ tel que, dans le cas où les termes croisés sont imposés nuls :

$$\left\| \begin{array}{ccc} \mathcal{F}_l(P, K)_{11} & & 0 \\ & \ddots & \\ 0 & & \mathcal{F}_l(P, K)_{NN} \end{array} \right\|_{\infty} < \gamma \quad (3.42)$$

Ceci revient à minimiser : $\max(\|\mathcal{F}_l(P, K)_{11}\|_{\infty}, \dots, \|\mathcal{F}_l(P, K)_{NN}\|_{\infty})$. Il est également possible d'utiliser cette propriété afin d'exécuter une synthèse \mathcal{H}_{∞} multi-objectifs, en imposant par exemple sur un autre canal séparé certaines contraintes au niveau du correcteur (roll-off via un filtre passe-haut par exemple).

Synthèse \mathcal{H}_{∞} multi-modèles

L'un des principaux intérêts de la synthèse \mathcal{H}_{∞} structurée telle qu'introduite dans ce chapitre concerne également la possibilité de considérer plusieurs réalisations LTI ($P_1(s), \dots, P_N(s)$) du système à commander, en même temps. Le schéma de principe est donné à la Figure 3.3.

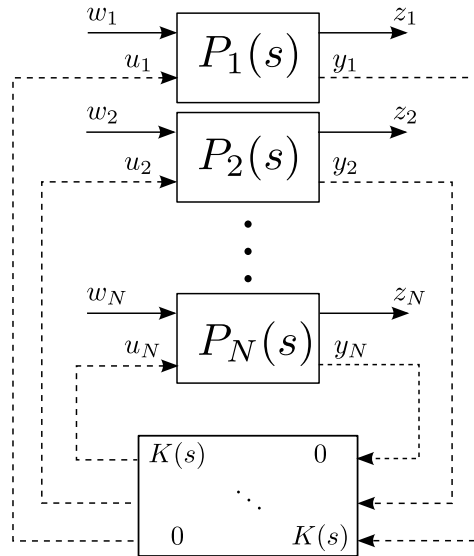


FIGURE 3.3 – Synthèse \mathcal{H}_{∞} multi-modèles.

Ceci a comme avantage principal de pouvoir considérer plusieurs configurations paramétriques dans le cas où le système initial dépend d'un vecteur de paramètres, afin d'assurer la robustesse de la loi externe vis-à-vis de ces configurations. Ces paramètres peuvent également inclure d'éventuelles incertitudes. Il est aussi possible d'introduire des pondérations plus fortes sur certains modèles afin d'accentuer l'importance que devra accorder le solveur à la minimisation des transferts sur ces réalisations précisément (afin d'accorder par exemple une importance accrue aux performances nominales).

Cette propriété sera fortement intéressante et plus particulièrement exploitée dans le chapitre 4, lorsque des modèles pire-cas issus de l'analyse de robustesse seront progressivement ajoutés à la synthèse.

CONCLUSION DU CHAPITRE

Ce chapitre a permis de donner quelques notions sur l'inversion dynamique non-linéaire. Cette technique possède l'avantage de pouvoir mêler une partie de commande non-linéaire ainsi que des techniques du monde linéaire, ces dernières pouvant notamment assurer une augmentation de la robustesse de l'ensemble. Du fait de son format et de la classe de systèmes qu'elle peut traiter, cette technique de compensation peut être particulièrement bien adaptée au pilotage robuste d'un avion sur un large domaine de fonctionnement. Une boucle interne de linéarisation permet de compenser les non-linéarités et d'adapter la loi au point de vol, et la boucle externe par synthèse \mathcal{H}_∞ structurée permet, par un correcteur d'ordre fixé, d'assurer les exigences en terme de dynamiques du système commandé.

Il existe cependant quelques points qui ont, jusqu'ici, été peu ou rarement abordés dans la littérature :

- la forme canonique obtenue après inversion dynamique, connue sous le nom de forme de Brunovsky, constituée d'une chaîne d'intégrateurs, peut poser des soucis de robustesse étant donné que toute la dynamique du système initiale n'est plus représentée,
- aucun moyen efficace, sans augmentation ni du conservatisme ni de la complexité du problème, a été développé afin de prendre en compte explicitement les éventuelles incertitudes structurées ou approximations du système.

Ces deux points constitueront les clés de voute de l'approche proposée au chapitre suivant, reprenant les bases de ce qui a été présenté dans ce chapitre introductif sur l'inversion dynamique robuste.

TECHNIQUE DE COMPENSATION NON-LINÉAIRE ROBUSTE

SOMMAIRE

4.1	DES SYSTÈMES AFFINES EN LA COMMANDE ET DÉPENDANT DE PARAMÈTRES	94
4.1.1	Description du système	94
4.1.2	Notations et hypothèses	95
4.2	UNE NOUVELLE APPROCHE BASÉE SUR L'INVERSION DYNAMIQUE ROBUSTE	96
4.2.1	Boucle interne de linéarisation	96
4.2.2	Boucle externe de robustification	97
4.3	MODÉLISATION LFT ET ANALYSE DE ROBUSTESSE	100
4.3.1	Incertitudes de modélisation	101
4.3.2	Incertitudes au niveau des actionneurs	101
4.3.3	Incertitudes sur les perturbations mesurées	102
4.3.4	Forme LFT standard du système bouclé	102
4.4	SYNTHÈSE MULTI-MODÈLES	103
4.5	VERS UNE ANALYSE GLOBALE	106
	CONCLUSION	108

Résumé

DANS ce chapitre est présentée une nouvelle technique de synthèse de loi de commande non-linéaire robuste basée sur l'inversion dynamique. Cette technique combine une boucle interne amenant le système d'origine à sa linéarisation Jacobienne, avec une boucle externe robuste rejetant les non-linéarités restantes, vues comme des perturbations mesurées. Cette dernière est dimensionnée en utilisant les dernières avancées en terme d'optimisation non-lisse, par synthèse \mathcal{H}_∞ structurée. La méthode ainsi développée inclue également un schéma d'analyse de robustesse qui permet d'extraire les configurations pire-cas, afin d'enrichir une banque de modèles. Les possibilités multi-modèles de la synthèse \mathcal{H}_∞ structurée sont alors exploitées au sein d'une procédure itérative de synthèse/analyse permettant d'améliorer les propriétés de robustesse de l'ensemble.

La technique de linéarisation entrée-sortie, introduite lors du précédent chapitre, permet à la fois de linéariser et de découpler une large classe de systèmes non-linéaires, et à la fois d'adapter automatiquement la loi de commande au point de fonctionnement. Ainsi elle constitue une alternative très intéressante notamment aux techniques de séquençement de gains, ou à d'autres méthodes de synthèse LPV qui, bien souvent, nécessitent des modèles à complexes et difficiles à élaborer [Marcos et Balas 2004].

Cependant les techniques d'inversion dynamique classiques sont souvent critiquées pour plusieurs raisons.

Tout d'abord, le problème le plus apparent est leur manque de robustesse face aux dynamiques négligées ou aux incertitudes de modélisation. La solution retenue dans ce cas est l'utilisation d'une loi de commande externe robuste, obtenue par exemple via une synthèse \mathcal{H}_∞ . Malgré cela, il est nécessaire d'évaluer la stabilité robuste de l'ensemble *a posteriori* [Bing-Yu et Blaise 1998; Papageorgiou et Glover 2004; 2005], ce qui généralement nécessite beaucoup de ressources et peut entraîner des ajustements par la suite, sans qu'aucune méthode ne permette réellement de pallier d'éventuels soucis de stabilité.

De plus, un autre problème non négligeable concerne l'absence de prise en compte de la dynamique des actionneurs. Cette dernière empêche en effet une compensation exacte des non-linéarités, ce qui peut entraîner une dégradation notable des performances de la loi. Il est donc important de considérer ces dynamiques lors de la phase de synthèse [Kara Mohamed et Lanzon 2013].

Enfin, et cette raison est un peu plus subtile, il a été observé notamment dans [Franco et al. 2006] que l'une des principales raisons pour laquelle les techniques d'inversion dynamique présentent des propriétés de robustesse assez pauvres concerne la forme trop générique obtenue, dite de Brunovsky, telle qu'exposée dans le chapitre précédent. Toute l'information de la dynamique du système d'origine est en effet perdue, ce qui empêche de la prendre explicitement en compte lors de la synthèse de la boucle externe. Il est alors proposé dans [Franco et al. 2006] de dimensionner la boucle interne linéarisante de façon à ce que le système résultant converge vers la linéarisation Jacobienne du système non-linéaire d'origine, autrement dit sa forme linéarisée autour d'un point d'équilibre donné. Ainsi, la synthèse de la boucle externe n'est plus basée sur un modèle générique mais bien sur un modèle dépendant explicitement des dynamiques linéarisées du système original.

Il est alors proposé dans ce chapitre une nouvelle méthodologie basée sur ces constats. Dans un premier temps à la section 4.1 est introduite la classe des systèmes

affines en la commande et dépendant de paramètres, sur laquelle la méthode de synthèse va être développée. On s'y intéresse particulièrement puisqu'elle est très représentative des systèmes d'aéronefs sur lesquels les applications seront portées. Puis à la section 4.2 sont exposées les étapes concernant la boucle interne de linéarisation d'une part, qui permet au système non-linéaire de se comporter comme sa linéarisation autour d'un voisinage, et la boucle externe de robustification d'autre part, qui vise notamment à rejeter les non-linéarités restantes pour étendre le domaine de fonctionnement, en plus d'assurer les performances du système bouclé. La section 4.3 introduit un moyen d'augmenter la robustesse de l'ensemble vis-à-vis d'incertitudes de modélisation, grâce à l'utilisation d'un schéma d'analyse de robustesse qui permet l'extraction de modèles pire-cas. Ces derniers sont exploités pour enrichir de façon itérative la banque de modèles de synthèse, sur laquelle un nouveau correcteur est dimensionné à chaque itération par synthèse \mathcal{H}_∞ multi-modèles et multi-objectifs. Un algorithme est proposé, constituant la clé de voute de l'ensemble de ces travaux, à la section 4.4. Enfin la section 4.5 vise à fournir quelques pistes supplémentaires pour améliorer le schéma d'analyse en prenant en compte plus explicitement la nature des perturbations et incertitudes.

4.1 DES SYSTÈMES AFFINES EN LA COMMANDE ET DÉPENDANT DE PARAMÈTRES

4.1.1 Description du système

Dans cette section on se propose de définir la classe de systèmes considérée, qui vise à décrire correctement des applications du monde aéronautique.

Soit le système à temps continu, affine en la commande, et dépendant de paramètres suivant :

$$\begin{cases} \dot{\xi}(t) &= f(\xi(t), \theta_p(t)) + G(\xi(t), \theta_p(t))u(t) \\ u(t) &= L_A(u_c(t)) \end{cases} \quad (4.1)$$

où $\xi(t) \in \mathbb{R}^n$ correspond aux états physiques du système évoluant au sein du domaine opérationnel $\mathcal{X} \subset \mathbb{R}^n$. La dynamique des actionneurs est traduite par l'opérateur linéaire à temps invariant et à gain unitaire $L_A(\cdot)$, qui convertit les entrées commandées $u_c(t) \in \mathbb{R}^m$ en entrées physiquement réalisées sur le système $u(t) \in \mathbb{R}^m$. Les non-linéarités ainsi que les variations paramétriques du système sont décrites par les fonctions $f(\cdot, \cdot) \in \mathbb{R}^n$ et $G(\cdot, \cdot) \in \mathbb{R}^{n \times m}$, qui dépendent toutes deux non seulement du vecteur d'états, mais également d'un ensemble de paramètres $\theta_p(t) \in \Theta \subset \mathbb{R}^r$. Ces paramètres peuvent représenter aussi bien des

configurations structurelles du système (masse, centrage, ...), que des variations de paramètres physiques (vitesse, altitude, nombre de Mach, ...).

Les notations et changements suivants sont introduits afin de faciliter l'application de la méthode, et notamment de la boucle interne partiellement linéarisante.

4.1.2 Notations et hypothèses

Soit $(\bar{\xi}, \bar{\theta}_p, \bar{u}) \in \mathbb{R}^n \times \mathbb{R}^r \times \mathbb{R}^m$ un point d'équilibre du système (4.1), qui satisfait donc la relation suivante :

$$f(\bar{\xi}, \bar{\theta}_p) + G(\bar{\xi}, \bar{\theta}_p)\bar{u} = \bar{f} + \bar{G}\bar{u} = 0 \quad (4.2)$$

En réécrivant $f(\xi(t), \theta_p(t))$ sous la forme :

$$f(\xi(t), \theta_p(t)) = \bar{f} + Ax(t) + \Delta_f(t) \quad (4.3)$$

avec :

$$A = \left. \frac{\partial f}{\partial \xi} \right|_{\bar{\xi}, \bar{\theta}_p} \quad \text{et} \quad x(t) = \xi(t) - \bar{\xi}(t) \quad (4.4)$$

on introduit ainsi sa linéarisation Jacobienne, où le terme supplémentaire $\Delta_f(t)$ représente l'écart entre la fonction non-linéaire $f(.,.)$ et son approximation linéaire.

Considérons maintenant l'hypothèse suivante dont la seconde relation (4.6) sera commentée dans la remarque 4.1.

Hypothèse 4.1

*Il existe une matrice constante $B \in \mathbb{R}^{n \times m}$ telle que, pour tout $(\xi(t), \theta_p(t)) \in \mathcal{X} \times \Theta$, une matrice **non singulière** $\Lambda \in \mathbb{R}^{m \times m}$ et une matrice d'erreur résiduelle $\Delta_G \in \mathbb{R}^{n \times m}$ peuvent être extraites telles que :*

$$G(\xi(t), \theta_p(t)) = B\Lambda + \Delta_G \quad (4.5)$$

$$\forall v(t) \in \mathbb{R}^m, L_A(\Lambda^{-1}v(t)) \approx \Lambda^{-1}L_A(v(t)) \quad (4.6)$$

Remarque 4.1

La matrice Λ peut être assimilée à l'efficacité de la commande. L'hypothèse concernant sa non-singularité n'est généralement pas restrictive pour des applications aéronautiques, notamment lorsqu'on considère le domaine d'évolution standard d'un aéronef lié à la notion de commandabilité du système. De plus, les termes diagonaux de la matrice varient lentement dans le temps, étant donné qu'ils dépendent principalement de la pression dynamique. Enfin, il arrive souvent que la matrice Λ ait ses termes diagonaux

numériquement prépondérants sur les autres, amenant à l'approximer par une matrice diagonale. Tout ceci justifie en pratique la propriété de commutativité de la relation (4.6) entre la matrice Λ^{-1} et la dynamique rapide des actionneurs représentée par l'opérateur $L_A(\cdot)$. Cependant un moyen de relâcher les contraintes de la relation (4.6) sera discuté dans la section 4.3.

4.2 UNE NOUVELLE APPROCHE BASÉE SUR L'INVERSION DYNAMIQUE ROBUSTE

4.2.1 Boucle interne de linéarisation

Il est proposé dans cette section d'appliquer une technique de commande basée sur l'inversion dynamique. Celle-ci permettra de compenser une partie des non-linéarités du système, tout en permettant à la loi externe de s'adapter automatiquement au point de fonctionnement.

Soient les deux signaux $v(t) \in \mathbb{R}^m$ et $\zeta(t) \in \mathbb{R}^m$. On définit la loi de commande intermédiaire linéarisante suivante :

$$u_c(t) = \Lambda(\zeta(t), \theta_p(t))^{-1}(v(t) - \zeta(t)) + \bar{u} \quad (4.7)$$

En combinant cette dernière avec l'équation du système (4.1), tout en considérant les notations de l'hypothèse 4.1, on obtient le système suivant :

$$\dot{x} = Ax + BL_A(v) + w_f + w_u \quad (4.8)$$

avec :

$$w_f = \Delta_f - BL_A(\zeta) \quad (4.9)$$

$$w_u = (G - \bar{G})\bar{u} + \Delta_G \Lambda^{-1} L_A(v - \zeta) \quad (4.10)$$

Il est important de noter que la dépendance en temps a été omise afin d'alléger les notations. Ainsi, grâce à la loi de commande (4.7), le système (4.1) a été transformé en un système linéaire représenté par l'équation (4.8), où :

- l'état x représente des faibles variations autour de la position d'équilibre considérée,
- les matrices A et B sont constantes, et correspondent donc aux matrices d'état du système linéarisé autour d'un point de fonctionnement donné,
- une nouvelle entrée, qui sera considérée comme la nouvelle commande du système, est représentée par v ,

- w_f et w_u , décrits par les équations (4.9) et (4.10), sont traités comme deux perturbations mesurées, et contiennent les non-linéarités restantes non compensées par la loi (4.7).

Remarque 4.2

Un des premiers intérêts de la méthode concerne la forme du système linéarisé (4.8). Contrairement aux méthodes classiques de linéarisation entrée-sortie, la forme obtenue n'est pas une série d'intégrateurs (communément appelée la forme de Brunovsky). Ceci permet de pallier les défauts inhérents à la forme classique, notamment son manque de robustesse puisque l'information des dynamiques du système initial est dans ce cas perdue. La loi de commande linéarisante (4.7) permet donc d'obtenir le modèle linéarisé autour d'un point de fonctionnement choisi, et si la nouvelle commande v rejette les perturbations mesurées w_f et w_u (qui contiennent les non-linéarités restantes), on peut s'attendre à obtenir une réponse dynamique qui reste assez proche du comportement linéarisé.

4.2.2 Boucle externe de robustification

L'objectif premier de la nouvelle commande v consiste donc à rejeter les perturbations w_f et w_u afin de garder le plus possible le comportement du système linéaire représenté par les matrices d'état A et B dans (4.8).

Hypothèse 4.2

En notant $w = w_f + w_u$, il est possible d'obtenir une mesure de w accessible à la commande. On note alors $\hat{w} = \hat{w}_f + \hat{w}_u$ l'estimation de ces perturbations obtenue.

Classiquement (voir la remarque 4.3 suivante), le terme w_f de l'équation (4.9) peut être partiellement compensé en utilisant un choix optimal du signal d'entrée $\zeta(t)$, de la façon suivante :

$$\hat{\zeta}(t) = \text{Arg} \min_{\zeta(t) \in \mathbb{R}^m} \|\Delta_f(t) - BL_A(\zeta(t))\| \quad (4.11)$$

La commande ζ doit être évaluée hors-ligne, et est en pratique souvent choisie très simplement de façon analytique, comme il sera montré dans le chapitre 6.

Remarque 4.3

Dans le cas spécial des systèmes carrés ($m = n$), avec une dynamique d'actionneurs parfaite ($L_A(u(t)) = u(t)$), on obtiendrait $w_f = 0$ en utilisant la commande $\hat{\zeta}(t) = B^{-1}\Delta_f(t)$. Ceci correspond typiquement à ce qui est fait dans un schéma d'inversion dynamique classique.

Afin maintenant de compenser les autres termes estimés contenus dans \hat{w} , au moins sur une bande limitée de fréquences, la nouvelle commande v peut être

utilisée. La structure de la boucle externe de commande suivante est considérée pour la suite, en introduisant le nouveau correcteur $K(s)$ qu'il faudra déterminer :

$$v = K(s) \begin{pmatrix} \hat{w} \\ w_c \\ y \end{pmatrix} \quad (4.12)$$

où w_c représente le vecteur de consignes à suivre du système, et y les signaux de mesures accessibles à la commande. On note z les sorties du système qui correspondent aux données à commander (le but étant que z suive les consignes w_c).

Hypothèse 4.3

Sans grande perte de généralités dans la plupart des applications réalistes, on suppose que y et z dépendent linéairement de l'état x , de façon à ce que $y(t) = Cx(t) \in \mathbb{R}^p$ et $z(t) = Lx(t) \in \mathbb{R}^q$.

Il reste maintenant à dimensionner la boucle externe, et plus particulièrement le correcteur $K(s)$. Il doit être obtenu de façon à ce que :

1. un bon suivi d'un modèle de référence soit réalisé, en minimisant l'erreur entre le signal de sortie z et le signal de référence $z_r = R(s)w_c$, où le modèle LTI $R(s)$ décrit la dynamique nominale de référence du système bouclé,
2. l'activité de commande soit raisonnable, afin que les saturations des actionneurs du système ne soient pas atteintes. Ceci est obtenu en limitant indirectement le signal d'entrée v ,
3. les perturbations estimées $\hat{w} = \hat{w}_f + \hat{w}_u$, qui n'ont pas pu être compensées par la loi linéarisante décrite par l'équation (4.7), soient rejetées, afin d'élargir le domaine d'action de la loi en s'assurant de rester le plus proche possible du comportement dynamique décrit par le système linéaire (4.8).

Remarque 4.4

Un quatrième point crucial sera également abordé au cours du développement de la méthode, afin que la loi de commande reste robuste face aux incertitudes de modélisation et aux éventuelles dynamiques négligées. Il sera traité en détails aux sections 4.3 et 4.4.

En notant $\Sigma_A(s)$ la matrice de transfert associée à l'opérateur linéaire $L_A(s)$ (représentant la dynamique des actionneurs), et $\Sigma(s)$ le système linéarisé de l'équation (4.8) :

$$\Sigma(s) = \begin{bmatrix} L \\ C \end{bmatrix} (sI - A)^{-1} \begin{bmatrix} I & B \end{bmatrix} \quad (4.13)$$

les 3 points précédents pour la synthèse du correcteur $K(s)$ peuvent être traduits en un problème de minimisation \mathcal{H}_∞ multi-objectifs. En se basant sur les notions abordées lors du chapitre 3, et en considérant l'interconnexion linéaire de la figure 4.1,

le correcteur $K(s)$ peut ainsi être évalué en résolvant le problème de minimisation suivant :

$$\hat{K}(s) = \underset{K(s) \in \mathcal{K}}{\operatorname{argmin}} \|\mathcal{T}_{w_c \rightarrow z_p}(s)\|_\infty \quad (4.14)$$

avec :

$$\begin{cases} \|\mathcal{T}_{w_c \rightarrow z_u}(s)\|_\infty \leq \gamma_u \\ \|\mathcal{T}_{w \rightarrow z_p}(s)\|_\infty \leq \gamma_r \end{cases} \quad (4.15)$$

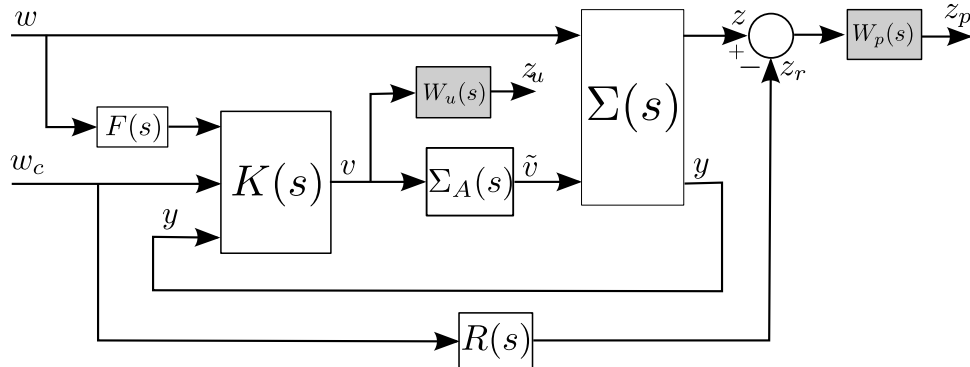


FIGURE 4.1 – Schéma de synthèse \mathcal{H}_∞ multi-objectifs.

Ainsi, la minimisation de la norme \mathcal{H}_∞ dans l'équation (4.14) correspond au critère de performance, permettant d'obtenir un bon suivi du modèle de référence $R(s)$. Comme décrit lors du chapitre 3, à la section 3.3, un filtre passe-bas $W_p(s)$ est introduit à ce propos afin de spécifier la plage de fréquences pour laquelle les performances de suivi doivent être accrues. Ceci permet de répondre au point 1 énuméré ci-dessus.

Les points 2 et 3 sont respectivement pris en compte par les deux contraintes additionnelles représentées en (4.15). La première implique notamment d'utiliser un filtre passe-haut $W_u(s)$ afin de limiter l'action de la commande à hautes fréquences, pour ainsi éviter une trop forte sollicitation des actionneurs qui amènerait à les saturer, d'abord en vitesse puis en amplitude. De telles saturations (en particulier les saturations de vitesse qui introduisent un déphasage) sont généralement à proscrire car elles vont induire des dégradations fortes de performance voire de stabilité du système. Quand elles ne peuvent être évitées, en raison d'un cahier des charges exigeant, on devra compléter le correcteur $\hat{K}(s)$ par un dispositif anti-windup [Biannic et Apkarian 2011].

Notons enfin la présence d'un filtre passe-bas $F(s)$ à gain statique unitaire dans le schéma 4.1. Ce filtre, moins classique, a une double fonction :

- il permet de limiter la plage de fréquences sur laquelle doit s'opérer la rejection des perturbations w , limitant ainsi les dynamiques du correcteur à des valeurs raisonnables,
- il traduit aussi le fait que les signaux w et \hat{w} ne coïncident que dans les basses

fréquences. Autrement dit, l'estimation \hat{w} utilisée par le correcteur n'est fiable que sur un domaine limité.

Afin de résoudre le problème d'optimisation (4.14)-(4.15) qui est non-convexe et non-lisse, les dernières avancées en optimisation non-lisse, développées notamment dans [Apkarian et Noll 2006; Gumussoy et Overton 2008], sont utilisées. Il est également possible de fixer à la fois la structure du correcteur $K(s)$, ce qui peut être intéressant dans de nombreux cas pratiques, ainsi que son ordre afin d'éviter des complications lors de l'implantation. Un autre point crucial de ces méthodes – qui visera à répondre notamment à la remarque 4.4 – est la possibilité de prendre en compte plusieurs modèles lors de la synthèse du correcteur. Ceci permet d'introduire encore plus de flexibilité afin de répondre aux objectifs et d'améliorer les propriétés de robustesse de la loi, notamment face aux incertitudes de modélisation. La section suivante aborde ce point de façon plus détaillée. Ce sera également l'opportunité de reprendre les notions abordées à la première partie (chapitres 1 et 2) de ce mémoire.

4.3 MODÉLISATION LFT ET ANALYSE DE ROBUSTESSE

Dans la section précédente, on a développé une méthode de synthèse de loi de commande fondée sur les techniques d'inversion dynamique robuste, avec les particularités suivantes :

- le modèle linéarisé obtenu n'est pas sous la forme canonique standard de Brunovsky, permettant de garder l'information sur la dynamique du système,
- les non-linéarités non-compensées sont considérées comme des perturbations mesurées ou estimées pouvant être rejetées par la boucle de commande externe afin d'élargir le domaine de fonctionnement,
- la boucle externe est synthétisée par synthèse \mathcal{H}_∞ structurée multi-objectifs, permettant d'assurer un bon suivi de modèle de référence, une commande limitée, ainsi qu'un rejet de perturbations.

Il reste cependant un point à aborder : comment valider la stabilité robuste de l'ensemble vis-à-vis des approximations introduites notamment lors de la synthèse ? Cette section développe ce point, en utilisant des outils d'analyse de robustesse des systèmes LTI, mettant en œuvre les techniques de μ -analyse développées dans la Partie I du manuscrit. Il existe en effet trois principales sources d'incertitudes.

4.3.1 Incertitudes de modélisation

En combinant les équations (4.7) et (4.1) afin d'obtenir (4.8), il a été supposé que les fonctions f et G étaient parfaitement connues. En réalité, issues d'une modélisation, elles contiennent des incertitudes, impliquant que la boucle interne de linéarisation n'a accès qu'à des estimations de ces fonctions. L'équation (4.8) devrait finalement s'écrire :

$$\dot{x} = A(\delta)x + B(\delta)L_A(v) + w \quad (4.16)$$

où $\delta = [\delta_1 \dots \delta_l]^T \in \mathbb{R}^l$ est un vecteur de paramètres incertains. Afin de pouvoir obtenir une LFT du système précédent, il est supposé sans grande perte de généralité que les matrices A et B dépendent de δ de façon rationnelle. Il est alors possible d'utiliser des techniques mentionnées lors du chapitre 1 (voir [SMAC 2012-2015; Roos et al. 2014]), afin de transformer le système $\Sigma(s)$ en la LFT suivante (représentée notamment à la figure 1.2) :

$$\Sigma(s) = \mathcal{F}_u(M(s), \Delta_M) \quad (4.17)$$

où $\mathcal{F}_u(\cdot)$ représente l'opérateur de la transformation linéaire fractionnaire supérieure (voir équation (1.7)), $M(s)$ est un système stable LTI, et :

$$\Delta_M \in \mathbf{\Delta}_M = \{diag(\delta_1 I_{k_1}, \dots, \delta_l I_{k_l}), \delta_i \in \mathbb{R}\} \quad (4.18)$$

4.3.2 Incertitudes au niveau des actionneurs

Afin d'obtenir le modèle linéarisé (4.8) (réécrit désormais sous la forme présentée en (4.16)), l'hypothèse 4.1 a du être employée afin d'assurer la commutativité entre l'opérateur $L_A(\cdot)$ et la matrice Λ^{-1} . En pratique cette hypothèse est loin d'être toujours vérifiée, et on se propose alors d'introduire la relaxation suivante.

Hypothèse 4.4

Il existe un opérateur non-linéaire borné $\Gamma(\cdot)$ et un réel positif $k_\Gamma \in \mathbb{R}^+$ tels que, $\forall v(t) \in \mathbb{R}^m$:

$$L_A(\Lambda^{-1}v(t)) = \Lambda^{-1}L_A(v(t)) + \Gamma(v(t)) \quad (4.19)$$

$$\|\Gamma(v(t))\|_{\mathcal{L}_2} \leq k_\Gamma \|v(t)\|_{\mathcal{L}_2} \quad (4.20)$$

Ainsi, en utilisant cette dernière hypothèse, il est possible de remplacer le modèle linéaire de l'actionneur $\Sigma_A(s)$ de la figure 4.1 par la version non-linéaire $\Sigma_A(s) + \Gamma(\cdot)$, comme il est montré dans la figure 4.2. Afin de se ramener à un paramètre LTI, quitte à rajouter un peu de conservatisme dans la formulation, la borne k_Γ pourra

être utilisée à la place de $\Gamma(\cdot)$. Ceci permet donc de prendre en compte par l'ajout d'une incertitude bornée l'éventuelle non-commutativité de $L_A(\cdot)$ avec Λ^{-1} .

4.3.3 Incertitudes sur les perturbations mesurées

Une dernière source d'incertitudes (du moins sur la modélisation approchée considérée dans cette étude), concerne les perturbations mesurées que la boucle externe doit rejeter. Ces perturbations représentent les termes non-linéaires que la boucle interne de linéarisation ne peut pas compenser. Initialement notées w , il est important de noter que seules des estimations de ces termes \hat{w} sont accessibles à la loi de commande externe $\hat{K}(s)$. Bien que lors de la synthèse de cette loi l'estimation est prise en compte grâce au filtre $F(s)$ (tel que $\hat{w} = F(s)w$, voir figure 4.1), cela peut paraître encore optimiste. Il est donc également important de vérifier la stabilité de l'ensemble en cas d'incertitudes d'estimation. Pour ce faire, un bloc diagonal Δ_w est considéré afin d'introduire des incertitudes multiplicatives sur le signal de façon à ce que \hat{w} devienne $(I + \Delta_w)\hat{w}$. Ce terme est une fois de plus visible sur le schéma d'analyse, figure 4.2.

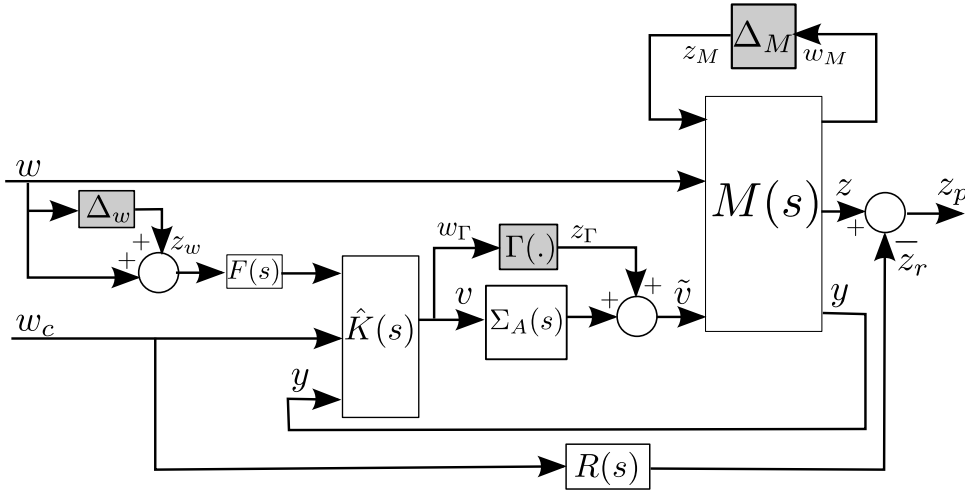


FIGURE 4.2 – Schéma d'analyse de stabilité robuste de la boucle fermée.

4.3.4 Forme LFT standard du système bouclé

Afin d'obtenir désormais la forme LFT standard pour appliquer des outils d'analyse de robustesse, les trois blocs d'incertitudes décrits ci-avant sont d'une part normalisés, et d'autre part regroupés afin de former une seule matrice Δ diagonale par blocs telle que :

$$\Delta = \text{diag}(\Delta_M, \Gamma(\cdot), \Delta_w) \quad (4.21)$$

Une fois isolé, le modèle linéaire augmenté correspondant $P(s)$ peut ainsi être généré, afin d'obtenir la forme standard représentée à la figure 4.3. Ces opérations peuvent être réalisées une fois de plus en utilisant la librairie APRICOT de la Toolbox Matlab SMAC (voir [SMAC 2012-2015]), en utilisant une banque de modèles LTI calculés pour diverses valeurs des incertitudes. Des techniques de μ -analyse, comme celles décrites dans la Partie I du manuscrit, sont alors utilisées afin d'évaluer des bornes inférieure $\underline{\mu}_\Delta$ et supérieure $\bar{\mu}_\Delta$ de la valeur singulière structurée sur toute la plage de fréquences. Puisque le bloc Δ est normalisé, le système est stable vis-à-vis d'incertitudes LTI si la borne supérieure $\bar{\mu}_\Delta$ vérifie $\bar{\mu}_\Delta < 1$. Sinon, il existe une perturbation déstabilisante pire-cas Δ^* telle que $\bar{\sigma}(\Delta^*) < 1$.

Remarque 4.5

Il faut noter que l'opérateur $\Gamma(\cdot)$, introduit à l'hypothèse 4.4, doit être remplacé par sa borne k_Γ afin que le bloc de perturbations Δ soit considéré LTI. Il sera proposé à la section 4.5 un moyen de limiter l'ajout d'un tel conservatisme dans la formulation.

Remarque 4.6

Bien entendu, grâce aux améliorations proposées dans le chapitre 2, l'écart entre les bornes inférieure et supérieure de la valeur singulière structurée pourra être maîtrisé et gardé relativement faible. Du fait que le bloc Δ ne contienne que des incertitudes paramétriques réelles possiblement répétées, un fort conservatisme peut être attendu dans certains cas en utilisant les méthodes classiques. Le risque ici d'un fort conservatisme de la borne supérieure est une évaluation garantie mais pessimiste de la marge de robustesse, évitant de garantir la stabilité alors qu'elle est en fait assurée. Ce point sera très important pour la méthode itérative proposée à la section suivante.

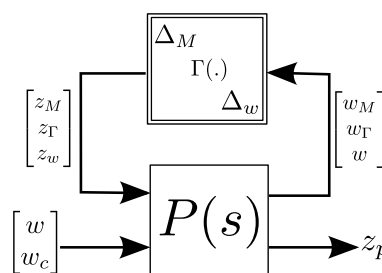


FIGURE 4.3 – Forme LFT du système bouclé pour l'analyse de robustesse.

4.4 SYNTHÈSE MULTI-MODÈLES

Une méthode pour analyser la stabilité robuste du système ayant été développée, il est proposé ici d'utiliser ces résultats pour la synthèse. Il peut exister pour ce faire deux méthodes.

Puisque le correcteur $\hat{K}(s)$ a été dimensionné au sein d'un schéma de synthèse \mathcal{H}_∞ , une des façons les plus naturelles consisterait à employer des techniques de μ -synthèse (voir par exemple [Doyle 1985; Balas et al. 1991; Young 1996]). Des outils existent déjà au sein de la Robust Control Toolbox de Matlab (voir [The MathWorks 2010]). Cependant, il y a deux principales limitations à l'utilisation de cette technique. La première est très similaire à celle rencontrée pour la μ -analyse : dans le cas d'incertitudes purement réelles et répétées, il existe des problèmes numériques lors de la résolution par (D,G)-K itérations. Seul le cas d'incertitudes complexes fonctionne convenablement. De plus, la deuxième limitation concerne un risque fort de conservatisme de la loi de commande obtenue. Si un tel correcteur est calculé, il permet de stabiliser toutes les configurations pour l'ensemble du domaine admissible d'incertitudes Δ . On obtient donc une condition suffisante de stabilité robuste. Néanmoins, est-il nécessaire de stabiliser tout l'ensemble du domaine d'incertitudes ? Un tel correcteur pourrait alors s'avérer en pratique certes robuste, mais probablement trop conservatif.

Une solution alternative consiste alors à utiliser une technique de synthèse multi-modèles. Celle-ci a tout d'abord été considérée par [Ackermann 1985], mais dans le cas de la synthèse \mathcal{H}_∞ elle amène des problèmes d'optimisation non-convexes. Néanmoins le chapitre 3 a montré qu'il était possible désormais de résoudre ce type de problèmes grâce aux dernières avancées en optimisation non-lisse, qui sont introduites au sein des outils de synthèse \mathcal{H}_∞ structurée. Ainsi et pour notre problème, un algorithme simple est proposé. A partir d'un modèle de synthèse nominal, une première loi de commande robuste est évaluée. La robustesse de celle-ci est testée en reprenant la méthode proposée lors de la précédente section. Dans le cas où le correcteur ne permet pas de stabiliser toutes les configurations admissibles, un pire-cas (associé à la matrice de perturbations Δ^*) est extrait et ajouté au premier modèle de synthèse. Un nouveau correcteur est synthétisé grâce aux possibilités multi-modèles de la synthèse \mathcal{H}_∞ structurée. La robustesse est à nouveau évaluée, et le principe général consiste donc à un schéma de synthèse itératif, enrichissant au fur et à mesure la banque de modèles afin d'obtenir à la fin un correcteur capable de stabiliser toutes les configurations admissibles. L'algorithme 5 résume la procédure.

Algorithme 5: Synthèse \mathcal{H}_∞ multi-modèles avec analyse de robustesse

Entrées : Modèle de synthèse nominal $\Sigma_0(s)$, en utilisant la loi interne linéarisante (4.7), menant à (4.8)

Sorties : Correcteur $\hat{K}(s)$

- 1 Initialiser $i = 0$;
- 2 Résoudre (4.14)-(4.15) afin de calculer un premier correcteur nominal $\hat{K}_0(s)$;
- 3 Calculer la forme LFT normalisée du problème en suivant la méthode expliquée en section 4.3, et particulièrement la sous-section 4.3.4 ;
- 4 Évaluer la stabilité robuste du système bouclé avec le correcteur $\hat{K}_0(s)$, en calculant $\underline{\mu}_\Delta$ et $\bar{\mu}_\Delta$ par μ -analyse (avec les améliorations proposées au chapitre 2) ;
- tant que** $\bar{\mu}_\Delta \geq 1$
 - 5 Extraire une perturbation pire-cas Δ_i^* telle que $\bar{\sigma}(\Delta_i^*) < 1$, en utilisant les outils proposés dans [SMAC 2012-2015]. Former le modèle de synthèse pire-cas correspondant $\Sigma_i(s)$;
 - 6 Enrichir la banque de modèles avec $\Sigma_i(s)$;
 - 7 Incrémenter $i \leftarrow i + 1$;
 - 8 Calculer $\hat{K}_i(s)$ par synthèse \mathcal{H}_∞ multi-modèles ;
 - 9 Calculer la forme LFT normalisée du problème avec le nouveau correcteur $\hat{K}_i(s)$;
 - 10 Évaluer la stabilité robuste du système bouclé correspondant et calculer $\underline{\mu}_\Delta$ et $\bar{\mu}_\Delta$ par μ -analyse ;
- 11 $\hat{K}(s) = \hat{K}_i(s)$ est un correcteur qui stabilise toutes les configurations admissibles ;

Remarque 4.7

Contrairement aux techniques de μ -synthèse, l'algorithme 5 amène des conditions uniquement nécessaires de stabilité robuste vis-à-vis de perturbations LTI, qui deviennent suffisantes une fois que la borne supérieure vérifie $\bar{\mu}_\Delta < 1$.

Remarque 4.8

Comme il a été mentionné au chapitre précédent à la sous-section 3.3.3, il est également possible d'introduire des pondérations sur les modèles de synthèse, et notamment une pondération plus forte sur le modèle nominal $\Sigma_0(s)$ afin de garder le plus possible les performances nominales. Par exemple, une pondération 5 fois plus importante est généralement un choix intéressant pour ne pas trop dégrader les performances du correcteur obtenu.

Il est important de souligner que l'arrêt de l'algorithme 5 est assuré uniquement lorsque la borne supérieure de la valeur singulière structurée est strictement

inférieure à 1. Or lorsqu'on est en présence d'incertitudes paramétriques réelles possiblement répétées (comme ce sera le cas dans nos applications aéronautiques), il arrive très souvent que la valeur de la borne supérieure $\bar{\mu}_\Delta$ soit erronée. En effet l'écart avec la valeur de la borne inférieure correspondante $\underline{\mu}_\Delta$ est dans ce cas très élevé, ce qui est généralement dû à un trop fort conservatisme de la borne supérieure (voir chapitre 1). Ainsi, dans le cas où $\underline{\mu}_\Delta < 1$ et $\bar{\mu}_\Delta > 1$, il peut être difficile de conclure. L'algorithme régénérerait un pire-cas à partir de là, augmentant ainsi le nombre d'itérations et ajoutant inutilement du conservatisme dans la nouvelle loi de commande obtenue.

C'est pourquoi il est fortement conseillé et intéressant d'utiliser les améliorations proposées au chapitre 2 afin de réduire cet écart entre les bornes pour s'assurer de leur précision. Il est notamment proposé :

- d'utiliser l'algorithme 2 pour calculer à chaque fois une borne supérieure par la formulation LMI partielle de la caractérisation par matrices (D,G), afin d'avoir une borne assez précise tout en n'ayant pas un temps de calcul excessif,
- d'utiliser l'algorithme 3 de branch-&-bound amélioré à chaque fois que l'écart entre les bornes est trop important et ne permet pas de conclure, ce qui amène souvent à obtenir une nouvelle borne supérieure proche de la borne inférieure et de valider la stabilité robuste de la loi, sans générer un nouveau modèle pire-cas. Les améliorations proposées à l'algorithme 3 permettent en outre de ne pas augmenter significativement le temps de calcul à cette étape.

4.5 VERS UNE ANALYSE GLOBALE

Jusqu'ici, une forte hypothèse simplificatrice a été introduite afin de pouvoir considérer un schéma d'analyse de robustesse LTI permettant d'employer des techniques de μ -analyse. Dans l'approche proposée à la section 4.3 du présent chapitre, le signal d'entrée non-linéaire w est traité en tant que perturbation externe à rejeter par la loi de commande. Il est pourtant clair, considérant les équations (4.9) et (4.10), que ce signal possède une structure propre qu'il est possible d'évaluer. Il dépend en effet très clairement de x et θ_p , mais d'une façon certes assez complexe. Il est difficile de considérer une telle dépendance lors de l'analyse, d'où le schéma simplifié mais finalement assez conservatif.

Pourtant, il serait raisonnable de considérer une dépendance assez simple de w en θ_p et x , une fois de bonnes connaissances acquises sur ses variations, de façon à obtenir une relation approchée de la forme suivante :

$$w = H(\theta_p)x + W(x) \quad (4.22)$$

où $H(\cdot)$ et $W(\cdot)$ sont des fonctions non-linéaires aux dimensions appropriées, respectant les deux hypothèses suivantes.

Hypothèse 4.5

$H(\cdot)$ dépend rationnellement de θ_p , permettant ainsi une expression sous la forme d'une LFT (voir chapitre 1).

Hypothèse 4.6

L'opérateur non-linéaire $W(x)$ satisfait aux conditions de Lipschitz.

Ainsi, la LFT $\mathcal{F}_u(P(s), \Delta)$ représentée à la figure 4.2, peut être modifiée en étant désormais notée $\mathcal{F}_u(\tilde{P}(s), \tilde{\Delta})$, de façon à ce que le nouveau bloc de perturbations $\tilde{\Delta}$ soit maintenant exprimé par :

$$\tilde{\Delta} = \text{diag}(\Delta_M, \Gamma(\cdot), \Delta_w, W(\cdot), \Theta_p) \quad (4.23)$$

afin de ne plus considérer w comme une perturbation externe. Il contient alors :

- des incertitudes paramétriques réelles Δ_w et Δ_M , qui sont des paramètres invariants dans le temps et sont issus d'une partie du vecteur θ_p ,
- des paramètres variants dans le temps Θ_p , comprenant également certains paramètres de θ_p qui avaient été supposés LTI alors qu'ils peuvent avoir une vitesse de variation non nulle (exemple de la masse ou de l'altitude sur un avion),
- des non-linéarités sans mémoire, éventuellement de secteur, $W(\cdot)$ et $\Gamma(\cdot)$, plutôt que de considérer leurs bornes qui amènent là encore un ajout non négligeable de conservatisme.

Le problème est qu'il n'est alors plus possible d'utiliser les outils de μ -analyse dans ce cas étendu, cette dernière n'étant valable que dans le cas de perturbations uniquement LTI. Il existe cependant un formalisme plus général pour analyser la stabilité robuste d'une telle configuration : le formalisme IQC (Contraintes Quadratiques Intégrales). Le lecteur pourra trouver dans [Megretski et Rantzer 1997; Köroğlu et Scherer 2007; Jönsson et Rantzer 1996; Scorletti 1997] quelques notions théoriques sur les IQC, et dans [Jönsson et al. 2004; Demourant 2013; Vuillemin et al. 2014] des considérations pratiques et algorithmiques. Bien que des améliorations aient été proposées récemment, l'un des principaux problèmes de l'utilisation des IQC est sa complexité numérique, qui entraînerait un temps de calcul assez prohibitif pour l'analyse de stabilité. Cependant, la reformulation du problème avec la réécriture de w selon l'équation (4.22), ainsi que la considération de non-linéarités de secteur et de paramètres variants dans le temps, peuvent être des pistes intéressantes à creuser à l'avenir afin d'améliorer sensiblement la phase d'analyse de robustesse et donc la détection de pire-cas pour la synthèse du correcteur.

CONCLUSION DU CHAPITRE

Inspirée des techniques d'inversion dynamique, la méthode proposée dans ce chapitre permet de synthétiser une loi de commande non-linéaire robuste adaptée à des systèmes ayant également des variations paramétriques. Elle utilise les principaux atouts des techniques d'inversion dynamique tels que la compensation des non-linéarités, le découplage, l'adaptation automatique de la loi au point de fonctionnement, et sa capacité à traiter une large classe de systèmes non-linéaires, tout en corrigeant ses principaux défauts que sont :

- l'obtention d'une forme trop générique composée d'une série d'intégrateurs, dite de Brunovsky, qui entraîne la perte des informations concernant la dynamique du système d'origine ; la structure de la boucle interne linéarisante permet en effet désormais d'aboutir à la linéarisation Jacobienne du système, avec les non-linéarités restantes considérées comme perturbations externes,
- le manque de robustesse vis-à-vis d'incertitudes de modélisation, en associant les possibilités de synthèse multi-modèles des récents algorithmes de synthèse \mathcal{H}_∞ structurée, à l'utilisation d'un schéma d'analyse de robustesse pour l'extraction de modèles pire-cas par μ -analyse,
- le manque de flexibilité quant à la gestion des systèmes non-carrés, sans avoir recours au calcul du degré relatif, tout simplement en sélectionnant les états pour lesquels les entrées ont le plus d'efficacité pour obtenir une matrice carrée généralement inversible Λ dans (4.5).

Cette stratégie de synthèse de loi se situe donc à la frontière des domaines de la commande non-linéaire et linéaire, tout en exploitant des outils puissants d'analyse de robustesse et les améliorations proposées au chapitre 2. Elle est de plus relativement aisée à mettre en œuvre, puisque la boucle interne de linéarisation est générique (lorsqu'un choix de la matrice Λ est réalisé, voir remarque 4.1), et qu'il ne reste ensuite plus qu'à exprimer le signal d'entrée ζ (souvent analytiquement en fonction de Δ_f), et à dimensionner le correcteur $K(s)$ de la commande v par synthèse \mathcal{H}_∞ réalisée sur le modèle linéarisé du système d'origine. La partie itérative avec la synthèse multi-modèles est facilement automatisable de telle sorte à ce que l'utilisateur ait peu de réglages à ajuster.

Bien que non restreintes à ce type d'applications, les techniques d'inversion dynamique classique sont particulièrement bien adaptées au monde des applications aéronautiques, et c'est d'ailleurs pour cette raison que ces travaux de thèse s'y sont focalisés, afin d'exploiter au mieux leurs avantages. La partie suivante vise donc à illustrer la méthode exposée uniquement de façon méthodologique et théorique dans ce chapitre, sur un problème de pilotage robuste d'un aéronef sur

l'ensemble de son domaine de fonctionnement, en présence d'incertitudes paramétriques. Avant d'entrer cependant dans l'application directe de la méthode de synthèse, quelques notions de mécanique de vol, essentielles pour mieux comprendre les aspects physiques des modèles considérés, seront introduits au premier chapitre de la partie suivante.

Troisième partie

Applications aéronautiques

ÉQUATIONS GÉNÉRALES DE LA MÉCANIQUE DU VOL

SOMMAIRE

5.1	NOTATIONS PHYSIQUES ET DÉFINITION DES REPÈRES	115
5.1.1	Notations physiques	115
5.1.2	Définition des repères	116
5.2	ÉQUATIONS GÉNÉRALES	119
5.2.1	Équations de la dynamique	119
5.2.2	Équations de la cinématique	120
5.3	FORCES ET MOMENTS	121
5.3.1	Expression des forces	121
5.3.2	Expression des moments	122
5.4	MODÈLE ATMOSPHÉRIQUE ET TURBULENCES	124
5.5	COEFFICIENTS AÉRODYNAMIQUES	126
	CONCLUSION	127

Résumé

QUELQUES notions de mécanique du vol sont rappelées dans ce chapitre, qui vise tout particulièrement à décrire les équations générales du mouvement d'un avion. Après avoir donné les principales notations ainsi que les définitions des différents repères utilisés, les équations de la dynamique et de la cinématique sont introduites afin d'obtenir un modèle à 12 états, caractéristique de ce genre de système. Puis les expressions des forces et moments sont détaillées, introduisant notamment la notion clé de coefficients aérodynamiques. Enfin quelques éléments sont apportés concernant la modélisation des perturbations atmosphériques et des turbulences.

L'obtention d'un modèle précis est capital en vue du pilotage automatique d'un avion. La mécanique du vol est la discipline qui étudie le mouvement des véhicules aériens, composé du déplacement d'un corps rigide et de la rotation de ce dernier autour de son centre de gravité. Elle s'appuie donc sur la mécanique du solide de Newton afin de décrire leur comportement dynamique et cinématique. Il existe cependant quelques particularités qui n'existent que pour les aéronefs : on peut notamment citer l'aérodynamique, qui permet de modéliser la génération de portance ou de traînée à partir d'une aile d'avion.

Avant d'explicitier clairement les équations générales de la mécanique du vol, la section 5.1 décrit dans un premier temps les diverses notations physiques utilisées pour le lecteur qui ne serait pas familier des applications aéronautiques. La section 5.2 présente quant à elle les équations de la cinématique et de la dynamique. Il en résulte le modèle dynamique classique d'un avion, exprimé par ses 12 variables d'état. Les sections 5.3 et 5.5 décrivent précisément l'expression des forces et des moments, ainsi que celle des coefficients aérodynamiques, qui apparaissent dans les équations générales. La section 5.4 rappelle également quelques notions sur les phénomènes de turbulences et propose un modèle atmosphérique simplifié, afin de prendre en compte des perturbations extérieures dans la modélisation.

5.1 NOTATIONS PHYSIQUES ET DÉFINITION DES REPÈRES

Cette section vise à introduire certaines notations qui seront utiles pour les deux chapitres qui suivront, ainsi que les principaux repères dans lesquels seront exprimées les équations de la mécanique du vol.

5.1.1 Notations physiques

Les équations générales de la dynamique d'un avion, qui sont présentées à la section 5.2, font appel à un grand nombre de notations, dont les principales sont présentées dans le tableau 5.1, et illustrées sur la figure 5.1. La plupart des données sont mesurables et accessibles au pilote et/ou à la loi de commande grâce à différents capteurs. Le pilotage longitudinal est réalisé par le braquage des gouvernes de profondeur δ_e et de poussée moteur δ_x . Le pilotage latéral est quant à lui composé d'une part du pilotage en roulis, réalisé par le braquage antisymétrique des ailerons δ_a , et d'autre part du pilotage en lacet, réalisé par le braquage de la gouverne de direction δ_r .

Notation	Description du paramètre	Unité
α	Angle d'incidence (ou d'attaque)	deg
β	Angle de dérapage	deg
δ_a	Angle de braquage des ailerons	deg
δ_e	Angle de braquage des gouvernes de profondeur	deg
δ_r	Angle de braquage de la gouverne de direction	deg
p	Vitesse de roulis	deg/s
q	Vitesse de tangage	deg/s
r	Vitesse de lacet	deg/s
γ	Pente	deg
ϕ	Angle de roulis (inclinaison)	deg
θ	Angle de tangage (assiette) $\theta = \alpha + \gamma$	deg
ψ	Angle de lacet (azimut)	deg
ρ	Masse volumique de l'air ($\rho_0 = 1.225 \text{ kg/m}^3$)	kg/m ³
V_a	Vitesse air	m/s
V_c	Vitesse calibrée (ou conventionnelle) $= \sqrt{\frac{\rho}{\rho_0}} V_a$	m/s
q_d	Pression dynamique $= \frac{1}{2} \rho V_a^2$	Pa

TABLEAU 5.1 – Liste des principales notations utilisées en mécanique du vol.

5.1.2 Définition des repères

Il existe plusieurs repères pour la modélisation du mouvement d'un avion (voir par exemple [Boiffier 1998] pour une description complète avec schémas). Cette sous-section vise à en introduire 3, ainsi que les matrices de changement de repère intervenant dans les équations de la mécanique du vol. Chacun d'eux trouve son utilité afin de simplifier l'expression des relations fondamentales. Tous les repères sont orthogonaux et à 3 dimensions, caractérisés par un centre, et 3 axes x_i , y_i et z_i .

Repère terrestre

Le repère terrestre, ou également appelé repère vertical, permet d'exprimer l'attitude de l'avion ainsi que la position par rapport à un autre repère inertiel. Il est défini par les caractéristiques suivantes :

- centre : centre de gravité G de l'avion,
- axe x_t : fixe, orienté vers le nord terrestre,
- axe y_t : fixe, orienté vers l'est.
- axe z_t : orthogonal à x_t et y_t , pointé vers la surface de la Terre,

Repère avion

Le repère avion est basé sur la géométrie de l'avion (il est représenté en bleu sur la figure 5.1). Ses caractéristiques sont les suivantes :

- centre : centre de gravité G de l'avion,
- axe x_b : orienté vers le nez de l'avion,

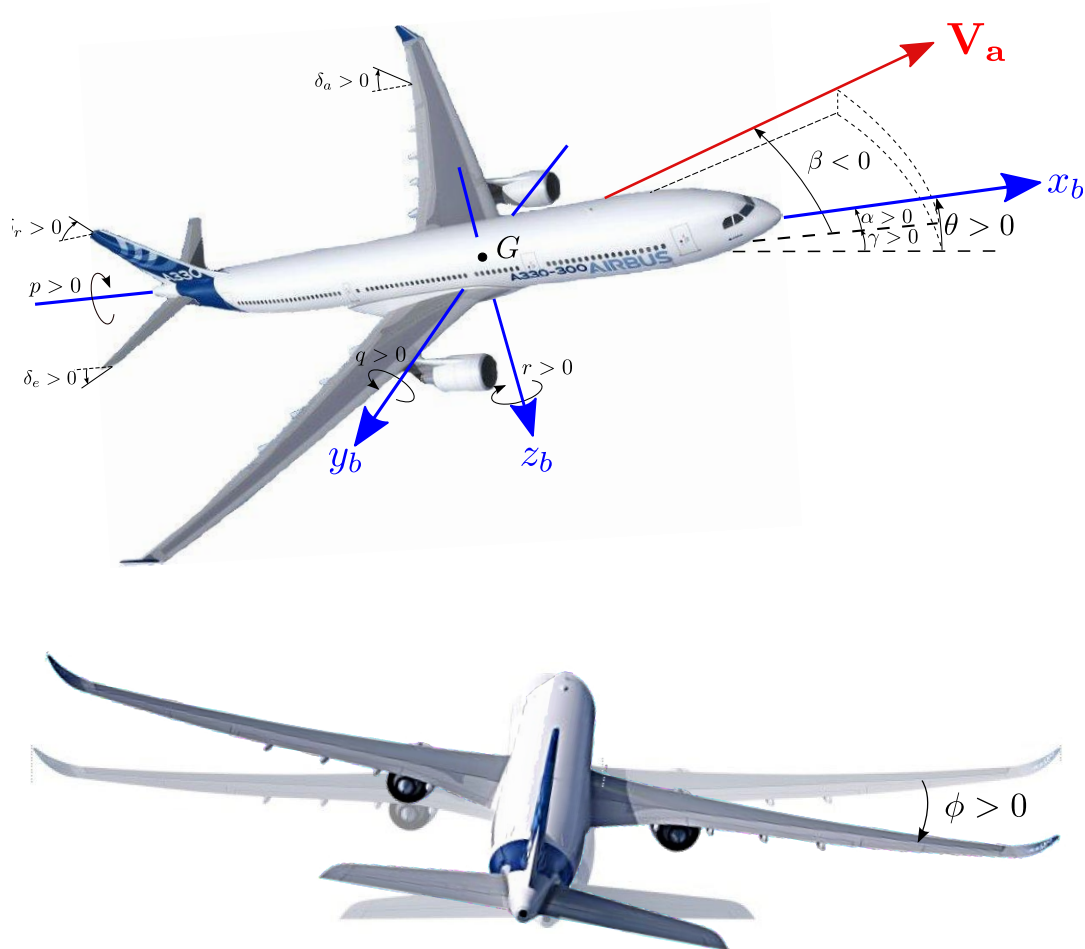


FIGURE 5.1 – Repère avion et description des diverses notations angulaires.

- axe z_b : orthogonal à x_b , dans le plan de symétrie de l'avion, et pointé vers la surface de la Terre,
- axe y_b : orthogonal à x_b et z_b , et dirigé vers l'aile droite.

Repère aérodynamique

Le repère aérodynamique est basé sur le vecteur vitesse \mathbf{V}_a . Ses caractéristiques sont les suivantes :

- centre : centre de gravité G de l'avion,
- axe x_a : orienté selon le vecteur vitesse \mathbf{V}_a ,
- axe z_a : orthogonal à x_a , dans le plan de symétrie de l'avion, et pointé vers la surface de la Terre,
- axe y_a : orthogonal à x_a et z_a , dirigé vers la droite.

C'est donc un repère non fixé puisque dépendant de l'orientation du vecteur vitesse. Il est utilisé pour décrire l'attitude de l'avion par rapport au vecteur vitesse.

Relations entre les repères

Quelques relations entre les repères, utiles pour la suite, sont données ici.

Du repère avion au repère terrestre Ce passage entre ces deux repères est utile lorsqu'il s'agit d'exprimer l'attitude de l'avion et ses vitesses de rotation. Le repère terrestre est obtenu à partir du repère avion par trois rotations d'angle $-\phi$ autour de l'axe x , puis d'angle $-\theta$ autour de l'axe y , et enfin d'angle $-\psi$ autour de l'axe z , décrites par les matrices :

$$R_x(-\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \quad (5.1)$$

$$R_y(-\theta) = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (5.2)$$

$$R_z(-\psi) = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.3)$$

La matrice de changement de repère est donc donnée par $R_{b \rightarrow t} = R_z(\psi).R_y(\theta).R_x(\phi)$, i.e. :

$$R_{b \rightarrow t} = \begin{bmatrix} \cos \theta \cos \psi & \sin \phi \sin \theta \cos \psi - \sin \psi \cos \phi & \sin \phi \sin \psi + \sin \theta \cos \phi \cos \psi \\ \cos \theta \sin \psi & \cos \phi \cos \psi + \sin \theta \sin \phi \sin \psi & \sin \theta \cos \phi \sin \psi - \sin \phi \cos \psi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \quad (5.4)$$

La matrice $R_{b \rightarrow t}$ étant orthogonale, le changement de base inverse est simplement donné par $R_{t \rightarrow b} = R_{b \rightarrow t}^T$.

Du repère aérodynamique au repère avion Le repère avion est obtenu à partir du repère aérodynamique par deux rotations d'angle $-\beta$ autour de l'axe z puis d'angle α autour de l'axe y , décrites par les matrices :

$$R_z(-\beta) = \begin{bmatrix} \cos \beta & \sin \beta & 0 \\ -\sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.5)$$

$$R_y(\alpha) = \begin{bmatrix} \cos \alpha & 0 & \sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \end{bmatrix} \quad (5.6)$$

La matrice de changement de repère est donc donnée par $R_{a \rightarrow b} = R_y(-\alpha).R_z(\beta)$, i.e. :

$$R_{a \rightarrow b} = \begin{bmatrix} \cos \alpha \cos \beta & -\cos \alpha \sin \beta & -\sin \alpha \\ \sin \beta & \cos \beta & 0 \\ \sin \alpha \cos \beta & -\sin \alpha \sin \beta & \cos \alpha \end{bmatrix} \quad (5.7)$$

Remarque 5.1

Une variante du repère aérodynamique existe, appelée "repère de stabilité". Ce dernier est centré en G , mais son axe x_s est orienté selon le projeté du vecteur vitesse \mathbf{V}_a sur le plan (G, x_b, z_b) . Il est donc obtenu à partir du repère avion par simple rotation d'angle $-\alpha$ autour de l'axe y_b . La matrice de changement de repère $R_{s \rightarrow b}$ pour passer du repère de stabilité au repère avion est donc définie par :

$$R_{s \rightarrow b} = \begin{bmatrix} \cos \alpha & 0 & -\sin \alpha \\ 0 & 1 & 0 \\ \sin \alpha & 0 & \cos \alpha \end{bmatrix} \quad (5.8)$$

Cette dernière sera notamment utilisée pour exprimer les forces aérodynamiques dans le repère avion, alors que les coefficients aérodynamiques sont généralement exprimés dans le repère de stabilité (voir section 5.3).

Il existe d'autres changements de repère, mais seuls ceux jugés utiles pour la suite ont été fournis dans cette section. Les équations générales peuvent maintenant être exposées à la prochaine section.

5.2 ÉQUATIONS GÉNÉRALES

Les équations générales du mouvement d'un avion sont données dans cette section, et sont extraites de [Boiffier 1998; Biannic et Roos 2015]. Le modèle dynamique décrit les interactions existantes entre les accélérations en translation et en rotation d'une part, et les diverses forces et moments d'autre part. Le modèle cinématique exprime quant à lui le mouvement de l'avion, en position et rotation, dans le repère terrestre. L'ensemble des relations dynamiques et cinématiques permettent d'obtenir les équations générales de la dynamique d'un aéronef.

5.2.1 Équations de la dynamique

Les vitesses de rotation et de translation sont notées respectivement :

$$\boldsymbol{\Omega} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (5.9)$$

et :

$$\mathbf{V} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (5.10)$$

Elles sont exprimées dans le repère avion. Les équations de la dynamique sont alors données par :

$$\dot{\mathbf{\Omega}} = I^{-1}(\mathbf{M} - \mathbf{\Omega} \times I \mathbf{\Omega}) \quad (5.11)$$

et :

$$\dot{\mathbf{V}} = \frac{\mathbf{F}}{m} - \mathbf{\Omega} \times \mathbf{V} \quad (5.12)$$

où m représente la masse de l'avion et I la matrice d'inertie exprimée par :

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \quad (5.13)$$

Les deux vecteurs \mathbf{F} et \mathbf{M} correspondent respectivement aux forces et aux moments. Ils sont tous deux exprimés dans le repère avion et leurs expressions seront détaillées à la section 5.3.

Remarque 5.2

Pour les systèmes aéronautiques, certains termes non diagonaux de la matrice d'inertie I donnée par l'équation (5.13) sont souvent négligeables par rapport aux termes diagonaux. Ceci permettra notamment de simplifier le problème de détermination de la matrice d'inertie d'un drone à voilure fixe au chapitre 7.

5.2.2 Équations de la cinématique

Les positions angulaires (angles d'Euler) et les coordonnées du centre de gravité G , exprimées dans le repère terrestre, sont notées respectivement :

$$\mathbf{\Phi} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \quad (5.14)$$

et :

$$\mathbf{X} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (5.15)$$

Les équations de la cinématique et de navigation sont alors données respectivement par les deux relations suivantes :

$$\dot{\Phi} = \underbrace{\begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{bmatrix}}_{T(\Phi)} \Omega \quad (5.16)$$

et :

$$\dot{\mathbf{X}} = \underbrace{\begin{bmatrix} \cos \theta \cos \psi & \sin \phi \sin \theta \cos \psi - \sin \psi \cos \phi & \sin \phi \sin \psi + \sin \theta \cos \phi \cos \psi \\ \cos \theta \sin \psi & \cos \phi \cos \psi + \sin \theta \sin \phi \sin \psi & \sin \theta \cos \phi \sin \psi - \sin \phi \cos \psi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix}}_{R_{b \rightarrow t}(\Phi)} \mathbf{V} \quad (5.17)$$

On obtient finalement un modèle à 12 états représentant complètement la dynamique et la cinématique de l'avion. Il est exprimé par :

$$\begin{cases} \dot{\mathbf{V}} &= \frac{1}{m} \mathbf{F} - \Omega \times \mathbf{V} \\ \dot{\Omega} &= I^{-1} (\mathbf{M} - \Omega \times I \Omega) \\ \dot{\mathbf{X}} &= R_{b \rightarrow v}(\Phi) \mathbf{V} \\ \dot{\Phi} &= T(\Phi) \Omega \end{cases} \quad (5.18)$$

Une fois ce modèle dynamique défini, il faut déterminer les expressions des forces et des moments. Cette partie est détaillée à la section suivante.

5.3 FORCES ET MOMENTS

5.3.1 Expression des forces

Les forces résultantes \mathbf{F} appliquées à un aéronef peuvent être décomposées en 3 termes : la poussée moteur \mathbf{F}_e , la gravité \mathbf{F}_g et les forces aérodynamiques \mathbf{F}_a . On obtient ainsi :

$$\mathbf{F} = \mathbf{F}_e + \mathbf{F}_g + \mathbf{F}_a \quad (5.19)$$

Il est important de noter que ces composantes doivent être exprimées dans le repère avion.

Poussée moteur

Il est supposé que la force de poussée \mathbf{F}_e est alignée avec l'axe longitudinal x_b , ce qui permet de simplifier les équations tout en restant très réaliste. On a donc :

$$\mathbf{F}_e = \begin{bmatrix} F_{e_x} \\ 0 \\ 0 \end{bmatrix} \quad (5.20)$$

Force de gravité

La force de gravité \mathbf{F}_g est alignée avec l'axe vertical z_t . Elle doit être exprimée dans le repère avion, d'où la nécessité de considérer la matrice de changement de repère $R_{t \rightarrow b}(\Phi)$:

$$\mathbf{F}_g = R_{t \rightarrow b}(\Phi) \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} = R_{b \rightarrow t}(\Phi)^T \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} = mg \begin{bmatrix} -\sin \theta \\ \cos \theta \sin \phi \\ \cos \theta \cos \phi \end{bmatrix} \quad (5.21)$$

où $g = 9.81 \text{ N/kg}$ représente l'accélération de la pesanteur.

Forces aérodynamiques

Les coefficients aérodynamiques sont généralement exprimés dans le repère de stabilité. Ainsi, afin de passer dans le repère avion, il faut effectuer une rotation d'angle α (voir remarque 5.1). L'expression des forces aérodynamiques est alors donnée par :

$$\mathbf{F}_a = \frac{1}{2} \rho V_a^2 S \underbrace{\begin{bmatrix} \cos \alpha & 0 & -\sin \alpha \\ 0 & 1 & 0 \\ \sin \alpha & 0 & \cos \alpha \end{bmatrix}}_{R_{s \rightarrow b}(\alpha)} \begin{bmatrix} C_X \\ C_Y \\ C_Z \end{bmatrix} \quad (5.22)$$

où S représente la surface de référence (qui dépend de l'application considérée). Les expressions des coefficients aérodynamiques de traînée $C_D = -C_X$, de force latérale C_Y et de portance $C_L = -C_Z$, seront détaillées à la section 5.5.

5.3.2 Expression des moments

Le moment total \mathbf{M} autour du centre de gravité G est dû au moteur \mathbf{M}_e et aux effets aérodynamiques \mathbf{M}_a , de sorte que :

$$\mathbf{M} = \mathbf{M}_e + \mathbf{M}_a \quad (5.23)$$

Ces deux composantes sont exprimées dans le repère avion.

Moment dû à la poussée moteur

Le point où s'exerce la poussée moteur, noté E , est différent du centre de gravité G . Il existe donc une contribution de celle-ci sur le moment total exercé au centre de gravité de l'avion, exprimée par :

$$\mathbf{M}_e = \mathbf{GE} \times \mathbf{F}_e = \begin{bmatrix} 0 \\ z_{GE} F_{e_x} \\ 0 \end{bmatrix} \quad (5.24)$$

avec \mathbf{GE} le vecteur reliant le centre de gravité G et le point d'application E de la poussée moteur \mathbf{F}_e , qui a comme coordonnées :

$$\mathbf{GE} = \begin{bmatrix} x_{GE} \\ 0 \\ z_{GE} \end{bmatrix} \quad (5.25)$$

Puisque la poussée moteur a été supposée alignée avec l'axe x_b du repère avion, le moment dû à la poussée moteur ne possède qu'une seule composante selon l'axe latéral y_b .

Remarque 5.3

Il apparaît clairement à l'équation (5.24) que la position verticale du moteur par rapport à celle du centre de gravité, autrement dit le signe de z_{GE} , entraîne soit un moment positif (à cabrer) soit un moment négatif (à piquer) de l'avion.

Moment dû aux forces aérodynamiques

Le moment dû aux forces aérodynamiques \mathbf{M}_a s'exprime par :

$$\mathbf{M}_a = \frac{1}{2} \rho V_a^2 S L \begin{bmatrix} C_l \\ C_m \\ C_n \end{bmatrix} + \mathbf{GA} \times \mathbf{F}_a \quad (5.26)$$

où :

- $L = \text{diag}(L_a, L_o, L_a)$ contient les longueurs de référence, avec :
 - L_a la longueur de référence latérale, qui peut être par exemple égale à l'envergure ou à la demi-envergure de l'aile,
 - L_o la longueur de référence longitudinale, souvent prise comme la corde aérodynamique moyenne,
- C_l , C_m et C_n sont les coefficients de moment respectivement autour des axes de roulis, de tangage et de lacet. Ils seront détaillés à la section 5.5, et leurs

expressions tout comme la façon de les obtenir définissent les conventions à utiliser pour les longueurs de référence L_a et L_o .

Le second terme de (5.26) provient du fait que les forces aérodynamiques ne s'appliquent généralement pas au centre de gravité G , mais au centre de poussée A . Empiriquement, ce dernier est généralement situé au tiers de la corde aérodynamique moyenne (en partant du bord d'attaque), mais ceci reste une approximation, et il n'est pas rare de négliger ce terme devant la contribution due aux coefficients de moment (terme de gauche de (5.26)).

Remarque 5.4

Il ne faut pas confondre le centre de poussée où s'appliquent les forces aérodynamiques avec le foyer aérodynamique qui correspond au point où le moment total ne varie pas quelque soit l'angle d'incidence α .

5.4 MODÈLE ATMOSPHERIQUE ET TURBULENCES

Les forces aérodynamiques et les moments dépendent de la pression dynamique $q_d = \frac{1}{2}\rho V_a^2$. Ils dépendent donc de la masse volumique de l'air, qui elle-même est fonction de l'altitude et de la température, mais également du vent, via la vitesse air.

La variation de la masse volumique de l'air ρ peut être modélisée assez simplement notamment à basse altitude. Les applications des deux prochains chapitres traitant respectivement de l'atterrissage automatique d'un avion de ligne et du pilotage d'un drone à voilure fixe, tous deux étant donc à des altitudes assez faibles, cette modélisation semble raisonnable. Ceci permet donc de négliger sans perte de précision les variations en fonction de l'altitude et de la température. La densité de l'air ρ dépend ainsi uniquement de l'altitude de la piste H_{pis} , de la température au niveau de la mer T_0 , et de la température au niveau de l'emplacement de la piste ou zone de départ/arrivée T_{pis} exprimée par :

$$T_{pis} = T_0 - 6.5 \times 10^{-1} H_{pis} \quad (5.27)$$

Ainsi, l'expression approximée de la variation de la masse volumique de l'air peut être donnée par :

$$\rho = \frac{353}{T_{pis}} \left(\frac{T_{pis}}{T_0} \right)^{5.25} \quad (5.28)$$

Le terme V_a correspond à la norme du vecteur de vitesse air, soit $V_a = \|\mathbf{V}_a\|$,

avec :

$$\mathbf{V}_a = \begin{bmatrix} V_{ax} \\ V_{ay} \\ V_{az} \end{bmatrix} \quad (5.29)$$

Ces composantes permettent notamment d'exprimer les angles d'attaque α et de dérapage β :

$$\alpha = \arctan \left(\frac{V_{az}}{V_{ax}} \right) \quad (5.30)$$

et

$$\beta = \arcsin \left(\frac{V_{ay}}{V_a} \right) \quad (5.31)$$

La vitesse air est généralement mesurée, mais il est important de comprendre sa provenance et ce qui l'influence. En notant \mathbf{V} le vecteur vitesse de l'avion, et \mathbf{W} le vecteur vitesse du vent exprimé dans le repère terrestre, la relation suivante est vérifiée dans le repère avion :

$$\mathbf{V}_a = \mathbf{V} - R_{t \rightarrow b}(\Phi) \mathbf{W} = \mathbf{V} - R_{b \rightarrow t}(\Phi)^T \mathbf{W} \quad (5.32)$$

La vitesse air est donc clairement affectée par le vent.

On peut décomposer le vecteur vitesse du vent \mathbf{W} en deux termes :

$$\mathbf{W} = \mathbf{W}_M + \mathbf{W}_T \quad (5.33)$$

dont le premier désigne le vent moyen tandis que le second correspond à des écarts induits par la turbulence atmosphérique. Il existe une modélisation largement reconnue pour cette dernière, utilisée lors des processus de certification : le modèle de Dryden. Étant donné que la turbulence est un phénomène stochastique défini par son spectre, il convient d'obtenir une modélisation fiable et représentative de celui-ci. Il est généralement donné selon les trois axes x_t , y_t et z_t respectivement afin de définir les trois composantes du vent turbulent, par les équations suivantes, fonctions de la fréquence ω :

$$\Phi_u(\omega) = \sigma_u^2 \frac{L_u}{\pi V} \frac{1}{(1 + (L_u \frac{\omega}{V})^2)} \quad (5.34)$$

$$\Phi_v(\omega) = \sigma_v^2 \frac{L_v}{\pi V} \frac{1 + 3 (L_v \frac{\omega}{V})^2}{(1 + (L_u \frac{\omega}{V})^2)^2} \quad (5.35)$$

$$\Phi_w(\omega) = \sigma_w^2 \frac{L_w}{\pi V} \frac{1 + 3 (L_w \frac{\omega}{V})^2}{(1 + (L_w \frac{\omega}{V})^2)^2} \quad (5.36)$$

où $\sigma_u, \sigma_v, \sigma_w, L_u, L_v$ et L_w sont des constantes, dont de nombreuses tables pour les choisir existent, notamment dépendant de l'altitude considérée.

5.5 COEFFICIENTS AÉRODYNAMIQUES

Dans les expressions des forces et moments aérodynamiques sont apparus divers coefficients aérodynamiques. Les coefficients de portance, de force latérale, et de traînée sont respectivement donnés par les expressions suivantes :

$$C_L = C_{L_0} + C_{L_\alpha} \alpha + C_{L_q} \frac{q}{V_a} + C_{L_{\delta_e}} \delta_e + \underbrace{C_{L_H} e^{-\lambda_L H_{LG}}}_{\text{effet de sol}} \quad (5.37)$$

$$C_Y = C_{Y_\beta} \beta + C_{Y_p} \frac{p}{V_a} + C_{Y_r} \frac{r}{V_a} + C_{Y_{\delta_a}} \delta_a + C_{Y_{\delta_r}} \delta_r \quad (5.38)$$

$$C_D = C_{D_0} + C_{D_\alpha} \alpha + C_{D_{\alpha^2}} \alpha^2 \quad (5.39)$$

dont les termes constants " C_{X_y} " sont appelés dérivées de stabilité, et sont déterminés par des essais en soufflerie ou par l'utilisation d'outils de soufflerie numérique, puis éventuellement recalés par des essais en vol et des procédés d'identification.

De façon similaire, les coefficients de moment autour des axes de roulis, tangage, et lacet sont donnés par :

$$C_l = C_{l_\beta} \beta + \frac{L_a}{V_a} \left(C_{l_p} p + (C_{l_{r_0}} + C_{l_{r_\alpha}} \alpha) r \right) + C_{l_{\delta_a}} \delta_a + C_{l_{\delta_r}} \delta_r \quad (5.40)$$

$$C_m = C_{m_0} + C_{m_\alpha} \alpha + \frac{L_0}{V_a} C_{m_q} q + C_{m_{\delta_e}} \delta_e + \underbrace{(C_{m_{H_0}} + C_{m_{H_\alpha}} \alpha) e^{-\lambda_m H_{LG}}}_{\text{effet de sol}} \quad (5.41)$$

$$C_n = (C_{n_{\beta_0}} + C_{n_{\beta_\alpha}} \alpha) \beta + \frac{L_a}{V_a} \left(C_{n_r} r + (C_{n_{p_0}} + C_{n_{p_\alpha}} \alpha) p \right) + C_{n_{\delta_a}} \delta_a + C_{n_{\delta_r}} \delta_r \quad (5.42)$$

Les derniers termes de (5.37) et (5.41) représentent l'effet de sol, qui devient actif lorsque l'avion se trouve à proximité de la piste. Ce phénomène devient assez conséquent lorsque la hauteur H_{LG} du train d'atterrissage par rapport au sol devient sensiblement égale à l'envergure de l'avion. Bien que n'étant pas dimensionnant lors d'un vol en palier à une altitude d'approche ou de croisière, il est important de le considérer au cours de la phase d'arrondi, qui sera notamment détaillée au chapitre 6 lors de l'étude de l'atterrissage automatique. Bien que difficile à modéliser et à comprendre, il entraîne une modification de la courbe de portance de l'engin, de façon à ce qu'à angle d'attaque fixé, un avion près du sol aura une

portance plus élevée. Ainsi, à portance égale, l'angle d'attaque d'un avion subissant l'effet de sol sera plus faible, limitant donc la traînée induite, et augmentant par ailleurs la finesse de l'appareil. Par conséquent, ce phénomène crée un effet de perturbation qui change le comportement dynamique de l'avion.

Remarque 5.5

Il est important de souligner que les expressions des coefficients aérodynamiques données précédemment sont des approximations au premier ordre, valables autour d'un point de fonctionnement donné. Elles restent valables sur l'ensemble du domaine de vol, mais les dérivées de stabilité deviennent alors des fonctions assez complexes de divers paramètres, tels le nombre de Mach et l'altitude. Étant données les applications considérées dans ce mémoire, les approximations (5.37)-(5.42) sont cependant largement satisfaisantes.

CONCLUSION DU CHAPITRE

Ce chapitre a présenté les équations générales de la mécanique du vol, qui permettent notamment d'établir un modèle précis d'un avion volant à relativement basse altitude (avec prise en compte de l'effet de sol). Ce modèle sera exploité dans les deux chapitres suivants, d'une part pour réaliser des objectifs de commande et de navigation pour un avion civil en phase d'atterrissage, et d'autre part afin de piloter automatiquement un drone à voilure fixe.

Il est proposé sur le site web de la boîte à outils Matlab SMAC [SMAC 2012-2015] un modèle Simulink qui regroupe l'ensemble des éléments détaillés dans ce chapitre, et qui peut servir à la fois pour la commande et pour la simulation. L'interface correspondante est présentée sur la figure 5.2 et est détaillée notamment dans [Biannic et Roos 2015].

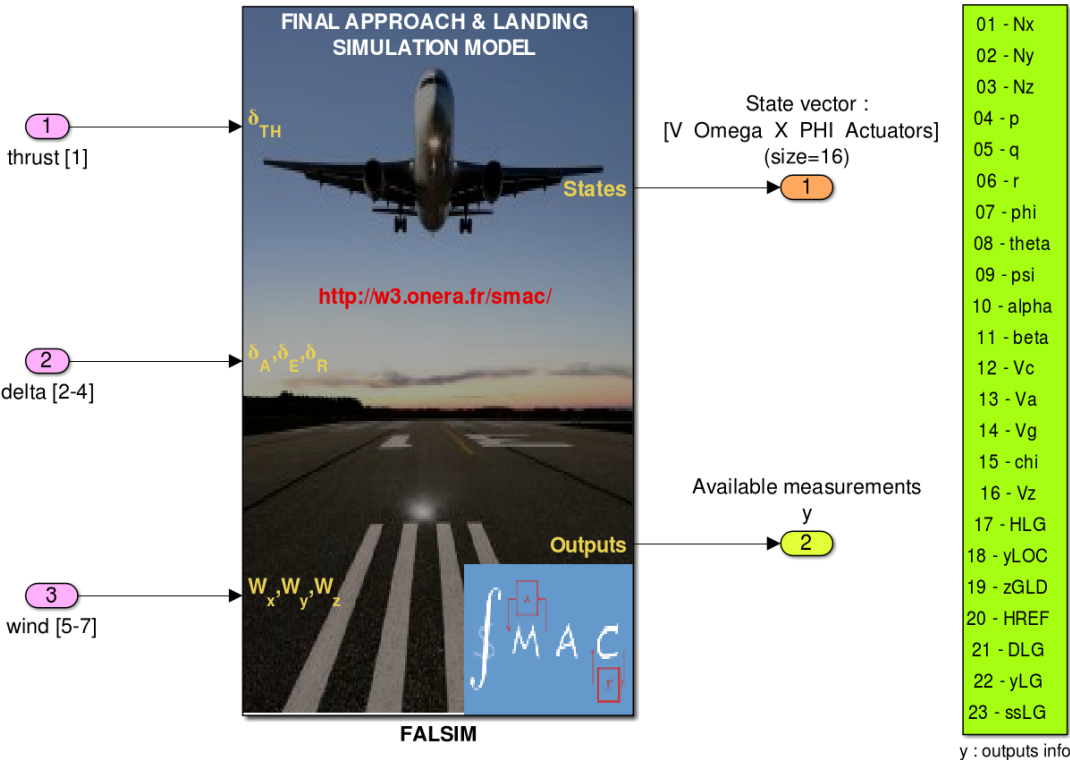


FIGURE 5.2 – Modèle Simulink générique d’un avion civil pour la simulation et la commande, présent dans la boîte à outils Matlab SMAC [SMAC 2012-2015].

APPLICATION À L'ATERRISSAGE AUTOMATIQUE D'UN AVION DE LIGNE

SOMMAIRE

6.1	SYNTHÈSE DU CORRECTEUR LONGITUDINAL	131
6.1.1	Modèle dynamique et objectifs	131
6.1.2	Application de la méthode de compensation non-linéaire	133
6.1.3	Analyse de robustesse et synthèse multi-modèles	136
6.2	SYNTHÈSE DU CORRECTEUR LATÉRAL	142
6.2.1	Modèle dynamique et objectifs	142
6.2.2	Application de la méthode complète et simulations	144
6.3	GUIDAGE POUR L'ATERRISSAGE AUTOMATIQUE PAR SUIVI D'ILS	146
6.3.1	ILS et guidage	147
6.3.2	Phases d'arrondi et de décroche	149
6.3.3	Simulations temporelles	150
	CONCLUSION	155

Résumé

LA méthode de compensation non-linéaire du chapitre 4, associant une boucle interne partiellement linéarisante (inspirée de l'inversion dynamique) à une boucle externe robuste dimensionnée par synthèse \mathcal{H}_∞ multi-modèles est mise en œuvre dans ce chapitre dédié au pilotage d'un avion civil en phase d'atterrissage. Des lois de pilotage longitudinales et latérales sont d'abord optimisées afin d'asservir pente et vitesse air d'une part, puis les angles de roulis et de dérapage d'autre part. Finalement, des lois de guidage assurant le suivi du faisceau ILS, puis l'arrondi et l'alignement sont déterminées afin de permettre un atterrissage complet jusqu'au toucher des roues du train principal malgré la présence d'incertitudes et de vent.

La méthode de synthèse de lois par compensation non-linéaire robuste, développée au chapitre 4, est appliquée ici pour le pilotage d'un avion civil (type Airbus A330) en phase d'atterrissage. Dans un premier temps la méthode complète permettra de synthétiser deux lois de commande à bas niveau afin de piloter en longitudinal la vitesse air V_a et pente γ , et en latéral les angles de roulis ϕ et de dérapage β . Ceci est détaillé aux sections 6.1 et 6.2, avec en plus des validations par des simulations temporelles sur le modèle non-linéaire complet, en présence de variations paramétriques et d'incertitudes de modélisation. La section 6.3 présentera la stratégie de guidage (boucle de commande plus haut niveau) afin de réaliser l'atterrissage automatique comprenant l'approche ainsi que la phase d'arrondi et l'éventuel décrochage en cas de vent latéral. Elle exposera également les notions principales concernant le guidage par ILS (Instrument Landing System), et des simulations temporelles pour vérifier si les diverses spécifications du cahier des charges sont tenues.

6.1 SYNTHÈSE DU CORRECTEUR LONGITUDINAL

6.1.1 Modèle dynamique et objectifs

Un modèle représentant la dynamique longitudinale d'un avion doit d'abord être déterminé. Plusieurs choix existent lorsqu'il s'agit d'exprimer des composantes longitudinales à partir des équations (5.18) présentées au chapitre précédent. Il est supposé tout d'abord que l'avion n'a aucune dynamique latérale active, autrement dit que la dynamique longitudinale est totalement découplée. Ceci impose donc $p = r = \beta = \phi = 0$. En exprimant la dynamique de la vitesse air V_a menant respectivement aux équations de force et de sustentation exprimées dans le repère aérodynamique, puis en gardant les relations des moments et cinématiques exprimées dans le repère avion, on arrive respectivement aux deux couples d'équations formant le modèle longitudinal à 4 états suivant (voir [Boiffier 1998], pages 131 – 133 pour de plus amples détails sur l'obtention de ce modèle) :

$$\begin{cases} m\dot{V}_a &= -q_d S C_D - mg \sin \gamma + F_{e_x} \cos \alpha \\ mV_a \dot{\gamma} &= q_d S C_L - mg \cos \gamma + F_{e_x} \sin \alpha \\ I_{yy} \dot{q} &= q_d S L_o C_m + z_{GE} F_{e_x} \\ \dot{\theta} &= q \end{cases} \quad (6.1)$$

où tous les paramètres ont été introduits au chapitre précédent, avec notamment l'expression des coefficients aérodynamiques C_D , C_L et C_m donnés respectivement aux équations (5.39), (5.37) et (5.41). On notera cependant \tilde{C}_L et \tilde{C}_m les parties respectivement de C_L et C_m qui ne contiennent pas l'entrée en braquage de gouverne

δ_e , soit :

$$\tilde{C}_L = C_{L_0} + C_{L_\alpha} \alpha + C_{L_q} \frac{q}{V_a} + C_{L_H} e^{-\lambda_L H_{LG}} \quad (6.2)$$

$$\tilde{C}_m = C_{m_0} + C_{m_\alpha} \alpha + \frac{L_o}{V_a} C_{m_q} q + (C_{m_{H_0}} + C_{m_{H_\alpha}} \alpha) e^{-\lambda_m H_{LG}} \quad (6.3)$$

Pour reprendre la formulation de la méthode de synthèse du chapitre 4, et notamment des équations (4.1)-(4.6), on a :

- le vecteur d'état $\xi = [V_a \quad \gamma \quad q \quad \theta]^T$, l'entrée $u = [F_{e_x} \quad \delta_e]^T$ et le vecteur de paramètres $\theta_p = [m \quad x_{cg} \quad V_{a0}]^T$ où m représente la masse de l'appareil, x_{cg} la position du centre de gravité G en pourcentage de la longueur totale du fuselage par rapport au nez de l'appareil sur l'axe x_b , et V_{a0} la vitesse air initiale qui variera lorsque des simulations seront faites à différents points de fonctionnement. Le domaine admissible Θ pour cette application est tel que $V_a \in [60 \ 90]$ m/s, $m \in [123 \ 180]$ tonnes, et $x_{cg} \in [15 \ 41]$ %,
- la dynamique des entrées de poussée moteur F_{e_x} et de braquage de gouverne de profondeur δ_e est modélisée par un système d'actionneurs du premier ordre $\Sigma_A(s)$ tel que :

$$\Sigma_A(s) = \begin{bmatrix} \frac{1}{2s+1} & 0 \\ 0 & \frac{1}{0.07s+1} \end{bmatrix} \quad (6.4)$$

de façon à ce que l'actionneur de la poussée ait une constante de temps de 2 secondes et celui du braquage de la profondeur de 0.07 secondes.

- les autres notations sont données par :

$$f = \begin{bmatrix} \frac{-q_d S C_D}{m} - g \sin \gamma \\ \frac{q_d S \tilde{C}_L}{m V_a} - \frac{g}{V_a} \cos \gamma \\ \frac{q_d S L_o}{I_{yy}} \tilde{C}_m \\ q \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (6.5)$$

$$\Lambda = \begin{bmatrix} \frac{\cos \alpha}{m} & 0 \\ \frac{z_{GE}}{I_{yy}} & \frac{q_d S L}{I_{yy}} C_{m_{\delta_e}} \end{bmatrix}, \quad \Delta_G = \begin{bmatrix} 0 & 0 \\ \frac{\sin \alpha}{m V_a} & \frac{q_d S}{m V_a} C_{L_{\delta_e}} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (6.6)$$

Remarque 6.1

En considérant le domaine admissible Θ donné précédemment pour cette application, il apparaît clairement que la matrice Λ est non-singulière, et donc inversible. De plus, les termes de Λ varient lentement puisqu'ils dépendent principalement de la pression dynamique q_d . Enfin, ses termes hors diagonaux sont numériquement très largement

négligeables devant ses termes diagonaux, rendant possible son approximation par une matrice diagonale. De ce fait, et étant donné que la matrice $\Sigma_A(s)$ est elle-même diagonale, toutes ces conditions vérifient bien l'hypothèse 4.1.

Remarque 6.2

Il est intéressant de discuter du choix de la matrice B , puisqu'il n'est pas unique, comme discuté lors de la méthode au chapitre 4. Ici, son choix a été fait en se basant sur l'efficacité maximale des commandes d'entrée F_{e_x} et δ_e sur les états du système. En effet, il apparaît nettement que la poussée F_{e_x} affecte tout particulièrement la dynamique de la vitesse air V_a , et que le braquage des gouvernes de profondeur δ_e agit largement dans un premier temps sur la taux de tangage q , celui-ci agissant ensuite sur l'angle de vol γ et sur le tangage θ (principe de séparation des échelles de temps, comme introduit lors de la sous-section 3.2.2 du chapitre 3).

Concernant enfin les objectifs de commande, il s'agira de piloter la vitesse air V_a et l'angle de pente (ou de vol) γ , en supposant que tout l'état soit accessible au correcteur. Le modèle de référence $R(s)$ est donné par :

$$R(s) = \begin{bmatrix} \frac{1}{6.5s+1} & 0 \\ 0 & \frac{0.35^2}{(2s+1)(s^2+2*0.7*0.35s+0.35^2)} \end{bmatrix} \quad (6.7)$$

et permet d'imposer une dynamique du premier ordre sur la vitesse et une dynamique du troisième ordre sur l'angle de vol. Ce choix est fait compte tenu de l'étude de la dynamique longitudinale d'un avion et des capacités des actionneurs sur ce modèle.

6.1.2 Application de la méthode de compensation non-linéaire

Maintenant que les notations du problème ont été introduites, il est possible d'appliquer la première étape de la méthode consistant à développer une loi interne linéarisante avec une première boucle externe visant à rejeter notamment les non-linéarités non compensées. Le problème de synthèse multi-modèles sera traité plus loin.

Il est d'abord considéré une configuration nominale, définie par :

$$\bar{\theta} = \begin{bmatrix} \bar{m} \\ \bar{x}_{cg} \\ \bar{V}_{a0} \end{bmatrix} \quad (6.8)$$

telle que $\bar{m} = 150$ tonnes, $\bar{x}_{cg} = 21\%$ et $\bar{V}_{a0} = 70$ m/s. Ce choix provient principalement du fait que cette configuration est "centrale" en ce qui concerne le placement

des pôles du système, permettant ainsi grâce aux propriétés de robustesse de la loi développée d'englober dès la première synthèse un certain nombre de configurations. A l'opposé, choisir comme configuration nominale une configuration dite "pire-cas" risquerait de fortement dégrader les performances sans assurer que les configurations opposées sur l'hypercube du domaine admissible soit stabilisables, ce qui ne constitue pas un choix a priori acceptable.

Concernant la première boucle interne de linéarisation, dont l'expression est donnée à l'équation (4.7) et permettant d'obtenir le modèle linéarisé (4.8), il faut dans un premier temps choisir la commande définie par $\hat{\zeta}$ permettant de minimiser l'impact de la perturbation mesurée w_f , tel qu'il est décrit par la condition (4.11). Un choix assez simple consiste à prendre $\hat{\zeta}$ dont les deux composantes sont respectivement égales à la première et la troisième ligne du vecteur Δ_f de (4.8), permettant ainsi de compenser la plupart des non-linéarités restantes pour les équations de la dynamique de V_a et q . Seules les termes non-linéaires de l'équation dynamique sur γ ne sont pas compensées par $\hat{\zeta}$. Ceux-ci seront traités par la boucle externe dont l'entrée est définie par v , avec le correcteur $K(s)$ introduit par la structure en (4.12). Ce dernier doit être synthétisé afin de répondre aux critères \mathcal{H}_∞ donnés par (4.14)-(4.15), avec le schéma de synthèse correspondant exposé à la figure 4.1. Il s'agit donc notamment de dimensionner les bonnes matrices de pondération $W_p(s)$ et $W_u(s)$. Un choix satisfaisant, répondant bien aux lignes directrices évoquées à la sous-section 3.3.2, est donné par :

$$W_p(s) = \begin{pmatrix} \frac{s/2+0.15}{s+1.5 \times 10^{-4}} & 0 \\ 0 & \frac{s/2+0.40}{s+4 \times 10^{-4}} \end{pmatrix} \quad (6.9)$$

$$W_u(s) = \begin{pmatrix} \frac{s+2}{0.001s+20} & 0 \\ 0 & \frac{s+0.2}{0.001s+2} \end{pmatrix} \quad (6.10)$$

Quant au filtre $F(s)$, il a été choisi comme suit :

$$F(s) = \begin{pmatrix} \frac{1}{2s+1} & 0 & 0 & 0 \\ 0 & \frac{1}{2s+1} & 0 & 0 \\ 0 & 0 & \frac{1}{2s+1} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (6.11)$$

où :

- le second terme diagonal de (6.11) peut posséder, pour le modèle de synthèse, une norme plus élevée que les autres de façon à ce que le correcteur cherche plus à se focaliser sur cette entrée que les autres. En effet, puisque le signal $\hat{\zeta}$ ne compense par les termes de l'équation en γ , il est important que la loi

de commande externe définie par $K(s)$ cherche à rejeter le plus possible les perturbations venant de cette partie,

- le dernier terme de la diagonale de (6.11) est nul puisque l'équation en $\dot{\theta}$ ne contient aucune non-linéarité.

Puisque grâce aux techniques avancées de synthèse \mathcal{H}_∞ structurée, il est possible de fixer l'ordre du correcteur $K(s)$, ce dernier est fixé à 3. En effet, ce choix présente l'avantage d'être un bon compromis entre les performances désirées à atteindre (par un ajout suffisant de degrés de libertés à la loi) et la complexité numérique, notamment dans le cas où il y aurait possibilité d'implanter sur le système réel le correcteur synthétisé (ce choix sera notamment crucial pour le dernier chapitre de ce mémoire, dédié au pilotage d'un drone à voilure fixe).

Le problème de minimisation multi-objectifs (4.14)-(4.15) est résolu et amène une norme \mathcal{H}_∞ optimale $\gamma_{opt} = 0.9653$, en utilisant la routine Matlab `hinfstruct`. Ceci permet d'affirmer que les spécifications fréquentielles sont bien respectées. Le correcteur nominal obtenu est noté $\hat{K}_0(s)$, et les figures 6.1 et 6.2 présentent les résultats de simulation réalisées sur le modèle non-linéaire complet.

Plus particulièrement, la figure 6.1 visualise les réponses de l'avion à une demande d'échelon de 3 degrés sur l'angle de vol γ , pour plusieurs configurations paramétriques telles que la masse $m \in \{123, 150, 180\}$ tonnes, les vitesses initiales $V_{a0} \in \{60, 70, 80\}$ m/s, et la position du centre de gravité varie telle que $x_{cg} \in \{15, 20, 40\}\%$.

De façon similaire, la figure 6.2 visualise les réponses à une variation de 3 m/s de la vitesse air. Dans les deux cas, la réponse du modèle de référence est présentée en tirets rouges, et est issue du modèle défini par $R(s)$ à l'équation (6.7).

Ces simulations mettent clairement en évidence l'efficacité de la méthode de compensation non-linéaire à assurer de bonnes performances ainsi qu'une certaine robustesse face aux variations paramétriques. Il est alors possible de considérer un domaine de fonctionnement assez large tout en gardant le plus possible les performances nominales, grâce notamment à la boucle interne linéarisante et à la boucle externe rejetant les non-linéarités restantes. De plus le découplage est également très bien assuré, étant donné que les variations de la vitesse air sont faibles lorsqu'un échelon est demandé sur l'angle de vol, et vice-versa. Cependant il reste un problème de taille qui n'a pas été traité jusqu'alors, qui constitue la dernière partie de la méthode. Aucune incertitude de modélisation n'a pour le moment été considérée, or il sera montré qu'elles ont malgré tout un grand impact sur la stabilité du système. Ceci est détaillé à la sous-section suivante.

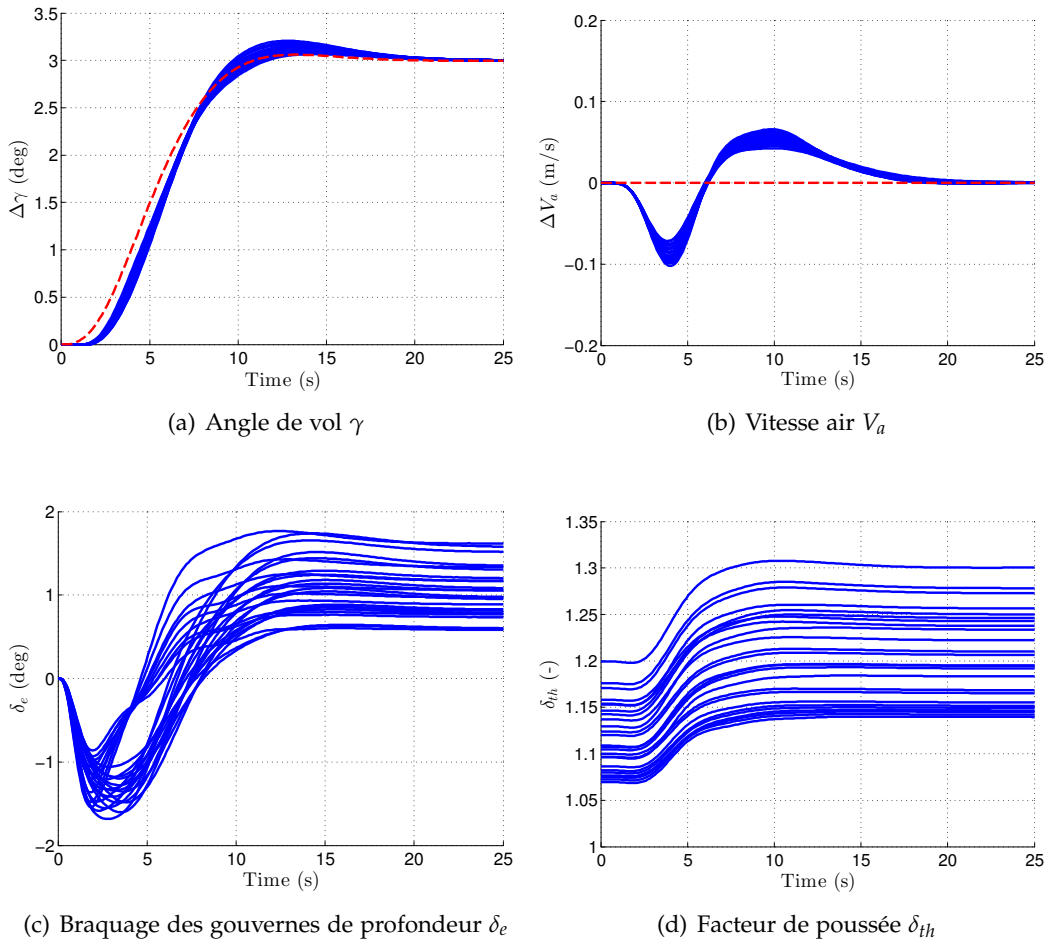


FIGURE 6.1 – Simulations sur le modèle non-linéaire complet, avec un échelon de 3 degrés sur l'angle de vol γ , pour différentes configurations paramétriques du domaine admissible Θ , et aucune incertitude de modélisation, avec le correcteur nominal $\hat{K}_0(s)$.

6.1.3 Analyse de robustesse et synthèse multi-modèles

Le correcteur $\hat{K}_0(s)$ synthétisé précédemment a été conçu afin d'assurer les performances de suivi du modèle de référence, d'avoir une activité de commande raisonnable, et également de rejeter les perturbations externes notamment afin d'élargir le domaine de fonctionnement. Cependant les opérateurs incertains – représentés par les boîtes "grisées" de la figure 4.2 du chapitre 4 – n'ont pas été considérés. Il faut donc dans un premier temps évaluer l'effet de ces incertitudes par l'utilisation d'outils de μ -analyse, et dans un second temps identifier les éventuels modèles pire-cas afin de les ajouter dans un processus de synthèse multi-modèles pour améliorer la robustesse globale de la loi.

Modèle LFT et μ -analyse

Tout d'abord, il faut établir la modélisation LFT du système incertain.

1. les incertitudes de modélisation au niveau des coefficients aérodynamiques

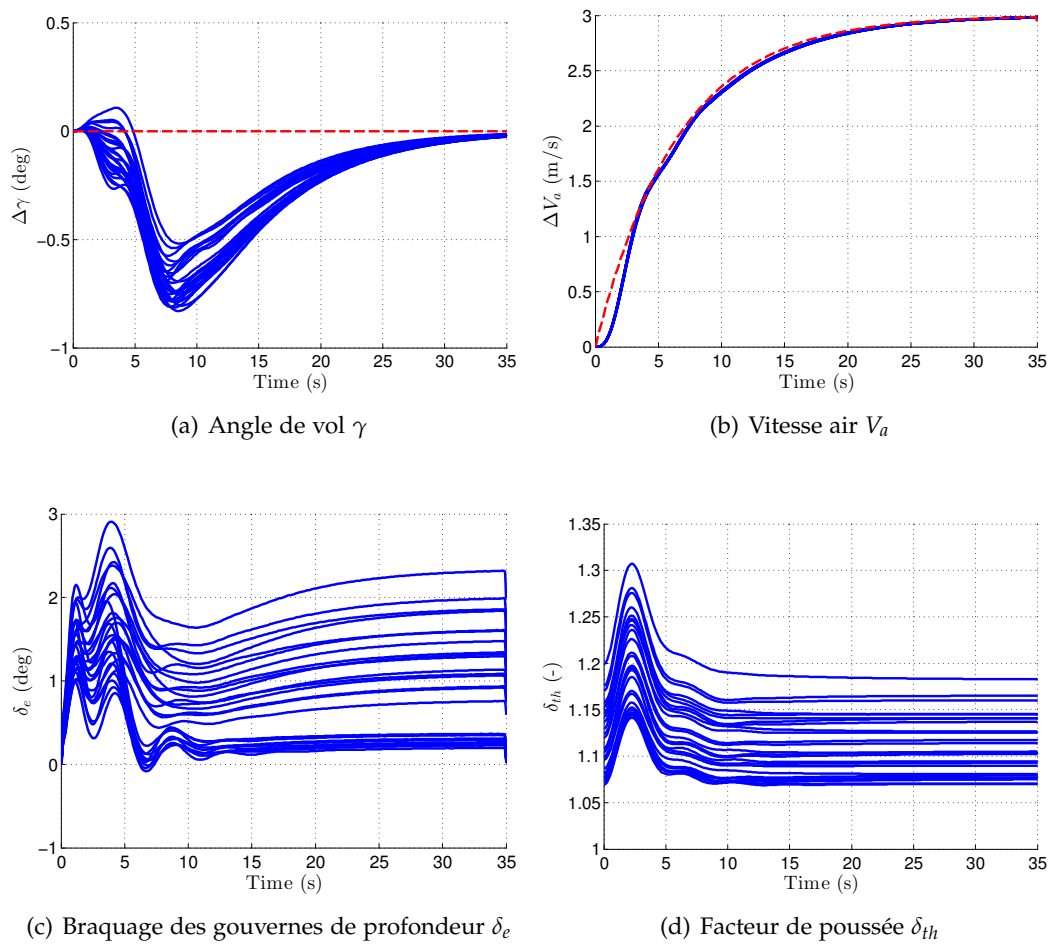


FIGURE 6.2 – Simulations sur le modèle non-linéaire complet, avec un échelon de 3 m/s sur la vitesse air V_a , pour différentes configurations paramétriques du domaine admissible Θ , et aucune incertitude de modélisation, avec le correcteur nominal $\hat{K}_0(s)$.

sont introduits, tels que les nouveaux coefficients aérodynamiques soient égaux à :

$$\hat{C}_D = (1 + \delta_{C_D})C_D \quad (6.12)$$

$$\hat{C}_L = (1 + \delta_{C_L})C_L \quad (6.13)$$

$$\hat{C}_m = (1 + \delta_{C_m})C_m \quad (6.14)$$

où δ_{C_D} , δ_{C_L} et δ_{C_m} sont des incertitudes multiplicatives normalisées,

2. les variations sur la vitesse air, la masse, et la position du centre de gravité sont introduites telles que $V_a = V_{a0} + \delta_V$, $m = m_0 + \delta_m$ et $x_{cg} = x_{cg0} + \delta_{x_{cg}}$.

Une série de modèles linéarisés selon l'équation (4.16) prenant en compte toutes ces variations est alors générée. En utilisant les outils présents dans la librairie APRICOT de la boîte à outils Matlab SMAC (voir [SMAC 2012-2015; Roos et al. 2014] et la section 1.1 du chapitre 1) une LFT d'ordre raisonnable $\mathcal{F}_u(M(s), \Delta_M)$

peut donc être obtenue à partir de cet ensemble de modèles LTI, avec :

$$\Delta_M = \text{diag}(\delta_V I_6, \delta_{C_D} I_2, \delta_{C_L} I_3, \delta_{C_m} I_2, \delta_m I_1, \delta_{x_{cg}} I_1) \in \mathbb{R}^{15 \times 15} \quad (6.15)$$

La LFT totale, incluant ensuite les blocs $\Gamma(\cdot)$ et Δ_w (voir figure 4.3), vérifie :

$$\Delta = \text{diag}(\Delta_M, \Delta_w, \Delta_\Gamma) \in \mathbb{R}^{20 \times 20} \quad (6.16)$$

où bien entendu, en vue de l'analyse de robustesse de systèmes LTI, l'opérateur $\Gamma(\cdot)$ est remplacé par l'opérateur LTI borné Δ_Γ (voir l'hypothèse 4.4).

Enfin une étape de normalisation est exécutée afin que les variations de Δ soient comprises dans la boule unité et incluent 30% d'incertitudes sur les coefficients aérodynamiques, et 30% de variations sur Δ_w et Δ_Γ . Il est alors possible d'utiliser les outils de μ -analyse afin d'évaluer une borne supérieure et une borne inférieure de la valeur singulière structurée, en utilisant les méthodes décrites aux chapitres 1 et 2. Dans un premier temps, avec le correcteur nominal $\hat{K}_0(s)$, une borne inférieure $\underline{\mu}_\Delta > 1$ est calculée, confirmant l'existence d'une configuration pire cas Δ_1^* telle que $\bar{\sigma}(\Delta_1^*) = \underline{\mu}_\Delta^{-1} < 1$. Cette configuration pire-cas correspond typiquement à une position arrière du centre de gravité, une masse maximale et des fortes valeurs sur les incertitudes des coefficients aérodynamiques.

Synthèse multi-modèles

Suivant l'algorithme 5, il faut alors procéder aux étapes 5 à 10 tant que la borne supérieure de μ_Δ n'est pas inférieure strictement à 1. Sur notre étude, 5 itérations supplémentaires sont nécessaires afin d'obtenir le correcteur multi-modèles final $\hat{K}(s)$ qui améliore sensiblement les propriétés de robustesse. Il faut cependant noter qu'un problème de conservatisme entre les bornes inférieure et supérieure est apparu, amenant $\underline{\mu}_\Delta < 1$ et $\bar{\mu}_\Delta > 1$. Grâce à l'utilisation de l'algorithme de branch-&-bound amélioré (voir algorithme 3 du chapitre 2 page 52), la borne supérieure peut être amenée très proche de la borne inférieure permettant d'obtenir $\bar{\mu}_\Delta < 1$. Le tableau 6.1 résume les valeurs des bornes obtenues avec le correcteur nominal $\hat{K}_0(s)$, puis avec le correcteur multi-modèles $\hat{K}(s)$, avant et après l'utilisation de l'algorithme de branch-&-bound.

Remarque 6.3

Il faut relever aussi que grâce aux améliorations proposées à l'algorithme 2 du chapitre 2 page 50, une borne supérieure de μ_Δ est à chaque itération évaluée rapidement et avec une grande précision par la caractérisation moins conservative LMI de la formulation par matrices (D, G) . Ceci montre l'intérêt des méthodes développées au chapitre 2.

Correcteur	μ -analyse		
	Méthode pour le calcul de $\bar{\mu}_\Delta$	$\underline{\mu}_\Delta$	$\bar{\mu}_\Delta$
$\hat{K}_0(s)$	Matrices (D,G), caractérisation LMI (Algorithme 2, page 50)	1.4610	1.5110
$\hat{K}(s)$	Matrices (D,G), caractérisation LMI (Algorithme 2, page 50)	0.8957	1.0780
	Branch-&-bound (Algorithme 3, page 52)	0.8999	0.9449

TABLEAU 6.1 – Valeurs des bornes inférieure et supérieure de μ_Δ pour l'analyse de stabilité robuste par μ -analyse, avec le correcteur nominal $\hat{K}_0(s)$ d'une part et le correcteur multi-modèles $\hat{K}(s)$ d'autre part.

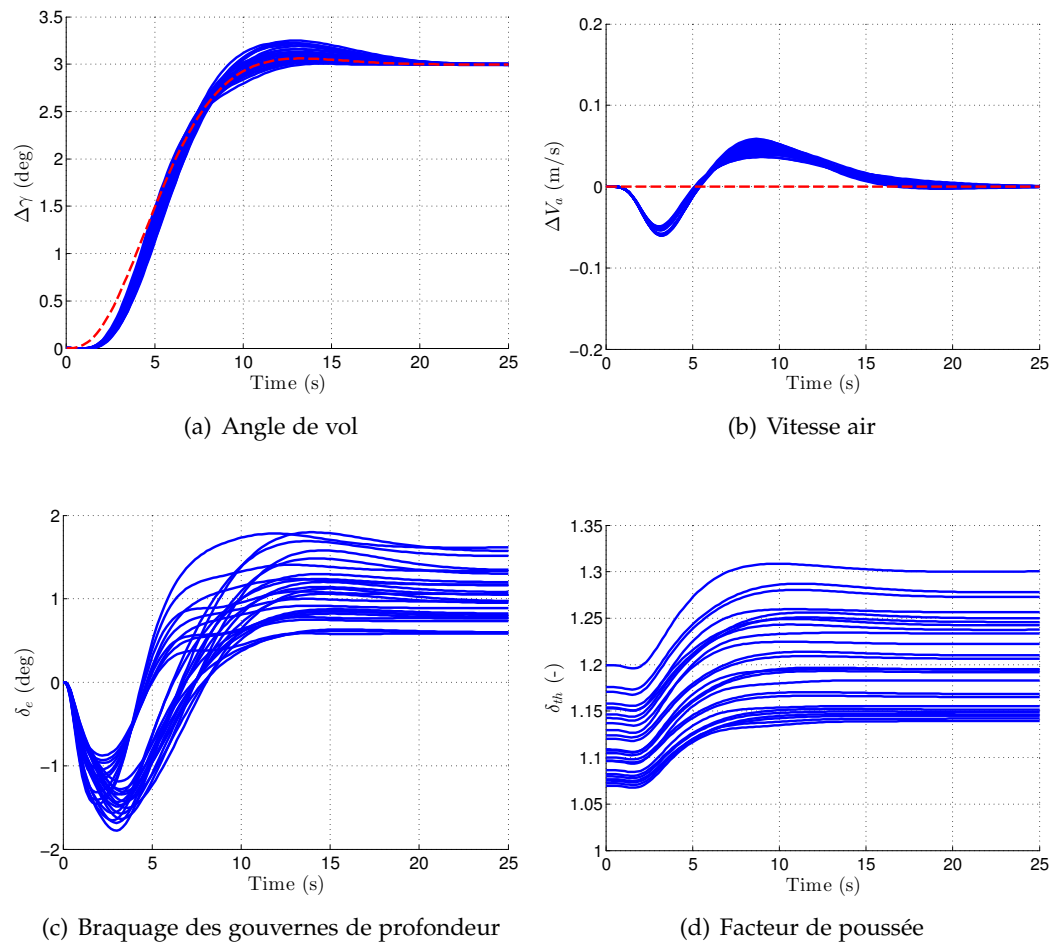


FIGURE 6.3 – Simulations sur le modèle non-linéaire complet, avec un échelon de 3 degrés sur l'angle de vol γ , pour différentes configurations paramétriques du domaine admissible Θ , et aucune incertitude de modélisation, avec le correcteur multi-modèles $\hat{K}(s)$.

Simulations temporelles

Avant de conclure cette section sur la synthèse du correcteur longitudinal avec la méthode développée au chapitre 4, quelques simulations temporelles sur le modèle non-linéaire complet sont réalisées. Comme pour le correcteur nominal, des sollicitations en angle de vol γ et en vitesse air V_a sont appliquées pour différentes configurations de vol (appartenant au domaine Θ), sans incertitudes de modélisation. Les résultats sont présentés aux figures 6.3 et 6.4, et peuvent être comparés à ceux obtenus avec le correcteur nominal $\hat{K}_0(s)$. Comme attendu, le découplage avec le correcteur $\hat{K}(s)$ est légèrement dégradé et le temps de réponse, notamment pour la vitesse air, est également plus lent. Cette perte de performance se veut toutefois très raisonnable, et il sera justifié par le gain en robustesse notamment vis-à-vis d'incertitudes de modélisation, grâce au processus multi-modèles.

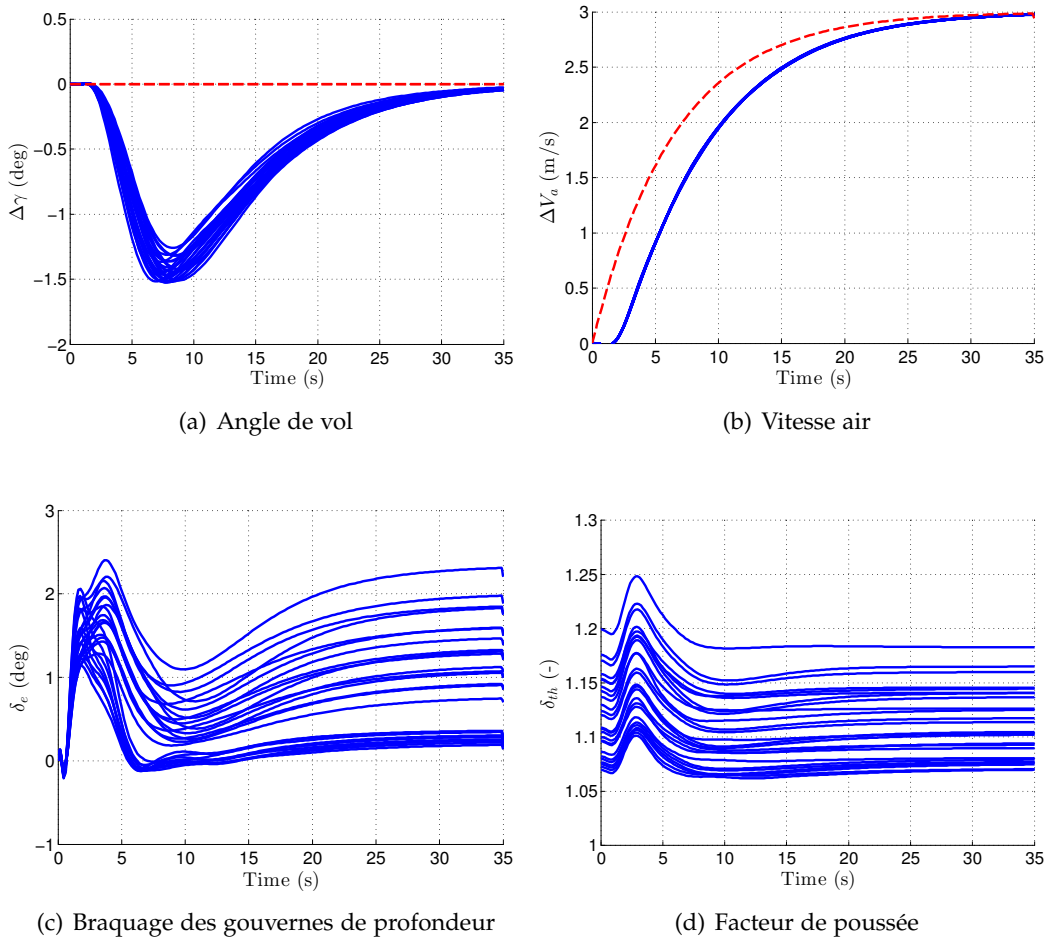
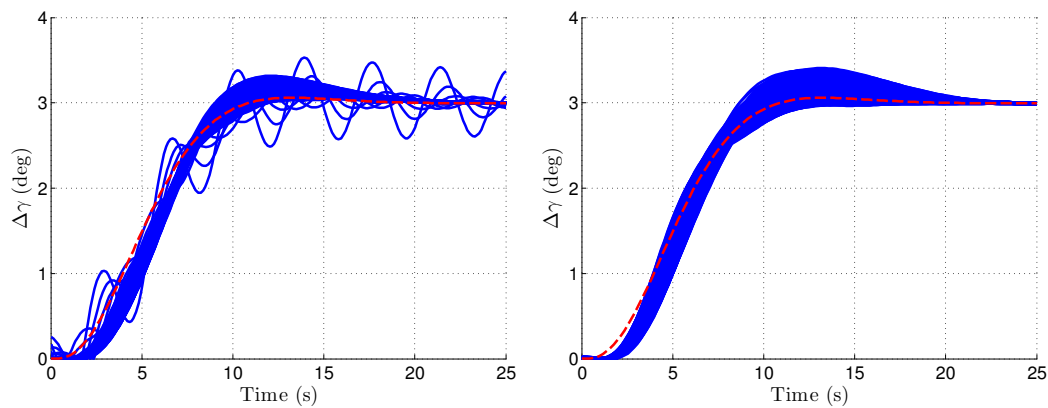


FIGURE 6.4 – Simulations sur le modèle non-linéaire complet, avec un échelon de 3 m/s sur la vitesse air V_a , pour différentes configurations paramétriques du domaine admissible Θ , et aucune incertitude de modélisation, avec le correcteur multi-modèles $\hat{K}(s)$.

Afin de souligner le bénéfice de cette synthèse multi-modèles, d'autres simulations ont été réalisées sur le suivi de l'angle de vol γ . Cette fois, 25% d'incertitudes

sur les coefficients aérodynamiques sont introduits (les 30% maximum déstabilisant complètement le correcteur nominal $\hat{K}_0(s)$, c'est pour cela que seulement 25% sont considérés ici). Les réponses temporelles pour un échelon en γ sont exposées sur la figure 6.5. La sous-figure de gauche (figure 6.5(a)) révèle bien le manque de robustesse du correcteur nominal $\hat{K}_0(s)$, corroborant l'analyse pire-cas résumée par la première ligne du tableau 6.1. Par contre, le correcteur multi-modèles $\hat{K}(s)$ présente clairement les améliorations notables d'avoir pris en compte quelques pire-cas, extraits de l'analyse, lors de la synthèse (figure 6.5(b)). Toutes les configurations sont stabilisées, et les performances restent largement dans des domaines acceptables. Ceci démontre clairement l'efficacité de la méthode.



(a) Sollicitations de 3 degrés en angle de vol γ , avec le correcteur nominal $\hat{K}_0(s)$.
 (b) Sollicitations de 3 degrés en angle de vol γ , avec le correcteur multi-modèles $\hat{K}(s)$.

FIGURE 6.5 – Comparaison entre le correcteur nominal $\hat{K}_0(s)$ (a) et le correcteur multi-modèles $\hat{K}(s)$ (b), pour toutes les configurations du domaine admissible Θ avec $\pm 25\%$ d'incertitudes de modélisation sur les coefficients aérodynamiques.

6.2 SYNTHÈSE DU CORRECTEUR LATÉRAL

Une fois le correcteur longitudinal synthétisé, la méthode est désormais appliquée à la synthèse du correcteur latéral, en utilisant la même stratégie que précédemment.

6.2.1 Modèle dynamique et objectifs

A partir des équations (5.18), et également des compléments apportés dans [Boiffier 1998], un modèle de la dynamique latérale est obtenu et est donné par :

$$\begin{cases} mV_a\dot{\beta} &= mV_a(p \sin \alpha - r \cos \alpha) + mg(\sin \theta \cos \alpha \sin \beta + \cos \theta \sin \phi \cos \beta \\ &\quad - \cos \theta \sin \alpha \cos \phi \sin \beta) + q_d S C_Y - F_{e_x} \sin \beta \cos \alpha \\ \dot{\phi} &= p + \tan \theta (q \sin \phi + r \cos \phi) \\ I_{zz}\dot{r} &= q_d S L_a C_n + (I_{xx} - I_{yy})pq \\ I_{xx}\dot{p} &= q_d S L_a C_l + (I_{yy} - I_{zz})rq \end{cases} \quad (6.17)$$

où les expressions des coefficients aérodynamiques C_Y , C_l et C_n sont données respectivement aux équations (5.38), (5.40) et (5.42). Comme pour les coefficients longitudinaux, on notera \tilde{C}_Y , \tilde{C}_l et \tilde{C}_n leurs parties respectivement qui ne contiennent pas les entrées d'ailerons δ_a ou de gouverne de direction δ_r , tel que l'on ait :

$$\tilde{C}_Y = C_{Y\beta}\beta + C_{Yp}\frac{p}{V_a} + C_{Yr}\frac{r}{V_a} \quad (6.18)$$

$$\tilde{C}_l = C_{l\beta}\beta + \frac{L_a}{V} \left(C_{lp}p + (C_{lr0} + C_{l_{r\alpha}}\alpha)r \right) \quad (6.19)$$

$$\tilde{C}_n = (C_{n_{\beta 0}} + C_{n_{\beta \alpha}})\beta + \frac{L_a}{V_a} \left(C_{nr}r + (C_{n_{p0}} + C_{n_{p\alpha}}\alpha)p \right) \quad (6.20)$$

Remarque 6.4

Il est intéressant de noter ici que des variables longitudinales sont présentes. Elles seront bien entendu considérées constantes lors de la synthèse du correcteur latéral. Pour les simulations, il faudra donc compter sur le correcteur longitudinal pour maintenir le plus possible les variables longitudinales à leur position d'équilibre.

Afin de reprendre le formalisme théorique de la méthode proposée au chapitre 4, on a les notations suivantes :

- le vecteur d'état $\xi = \begin{bmatrix} \beta & \phi & r & p \end{bmatrix}^T$, l'entrée $u = \begin{bmatrix} \delta_a & \delta_r \end{bmatrix}^T$ et le vecteur de paramètres (inchangé) $\theta_p = \begin{bmatrix} m & x_{cg} & V_{a0} \end{bmatrix}^T$,
- la dynamique actionneur des entrées δ_a et δ_r est modélisée chacune par un

premier ordre, tel que l'on ait :

$$\Sigma_A(s) = \begin{bmatrix} \frac{1}{0.06s+1} & 0 \\ 0 & \frac{1}{0.2s+1} \end{bmatrix} \quad (6.21)$$

– les autres notations sont données par :

$$f = \begin{bmatrix} p \sin \alpha - r \cos \alpha + \frac{S}{V_a} (\sin \theta \cos \alpha \sin \beta + \cos \theta \sin \phi \cos \beta - \cos \theta \sin \alpha \cos \phi \sin \beta) + \frac{q_d S}{m V_a} \tilde{C}_Y - \frac{F_{cx}}{V_a} \sin \beta \cos \alpha \\ p + \tan \theta (q \sin \phi + r \cos \phi) \\ \frac{q_d S L_a}{I_{zz}} \tilde{C}_n + \frac{I_{xx} - I_{yy}}{I_{zz}} p q \\ \frac{q_p S L_a}{I_{xx}} \tilde{C}_l + \frac{I_{yy} - I_{zz}}{I_{xx}} r q \end{bmatrix} \quad (6.22)$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (6.23)$$

$$\Lambda = \begin{bmatrix} \frac{q_d S L_a}{I_{zz}} C_{n_{\delta_a}} & \frac{q_d S L_a}{I_{zz}} C_{n_{\delta_r}} \\ \frac{q_d S L_a}{I_{xx}} C_{l_{\delta_a}} & \frac{q_d S L_a}{I_{xx}} C_{l_{\delta_r}} \end{bmatrix} \quad (6.24)$$

$$\Delta_G = \begin{bmatrix} \frac{q_d S}{m V_a} C_{Y_{\delta_a}} & \frac{q_d S}{m V_a} C_{Y_{\delta_r}} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (6.25)$$

Une fois encore, la matrice Λ est parfaitement inversible et sa variation ne dépend que de la pression dynamique q_d , permettant de satisfaire aux hypothèses 4.1. La matrice B a été choisie ici pour se concentrer sur les dynamiques rapides du système, soit les vitesses angulaire de roulis et de lacet p et r , pour lesquelles les commandes δ_a et δ_r respectivement sont les plus efficaces. Ce procédé s'inscrit de plus dans une logique de séparation des échelles de temps, afin de contrôler via la boucle interne les dynamiques rapides du système.

Concernant les objectifs de commande, il s'agira de piloter l'angle de roulis ϕ et l'angle de dérapage β , selon un modèle de référence $R(s)$ donné par :

$$R(s) = \begin{bmatrix} \frac{1}{\frac{s^2}{0.80^2} + \frac{2*0.75}{0.80}s + 1} & 0 \\ 0 & \frac{1}{\frac{s^2}{0.75^2} + \frac{2*0.75}{0.75}s + 1} \end{bmatrix} \quad (6.26)$$

permettant d'imposer des dynamiques du second ordre sur chacun, ce qui est cohérent avec la dynamique naturelle de l'avion.

6.2.2 Application de la méthode complète et simulations

La méthode de synthèse complète, incluant également le processus multimodèles, est désormais appliquée afin de synthétiser le correcteur latéral. Dans cette section, on ne détaillera pas autant la méthode que pour le cas longitudinal, mais on présentera les principales étapes.

Tout d'abord, la boucle interne de linéarisation est facilement introduite en reprenant les notations précédentes. Il reste uniquement à préciser le choix pour le vecteur $\hat{\zeta}$, dont les deux composantes sont égales aux 3^{ème} et 4^{ème} lignes de Δ_f , permettant de compenser la plupart des non-linéarités restantes pour les équations des dynamiques de r et p respectivement.

Concernant la synthèse de la boucle externe, on veillera ici à indiquer les pondérations pour le schéma de synthèse \mathcal{H}_∞ choisies. Un choix satisfaisant pour $W_p(s)$ et $W_u(s)$ est donné par :

$$W_p(s) = \begin{bmatrix} \frac{s/2+2.5}{s+2.5 \times 10^{-3}} & 0 \\ 0 & \frac{s/2+1}{s+1 \times 10^{-3}} \end{bmatrix} \quad (6.27)$$

$$W_u(s) = \begin{bmatrix} \frac{s+2}{0.001s+20} & 0 \\ 0 & \frac{s+0.5}{0.001s+5} \end{bmatrix} \quad (6.28)$$

Et le filtre $F(s)$ a été choisi tel que :

$$F(s) = \begin{bmatrix} \frac{1}{s+1} & 0 & 0 & 0 \\ 0 & \frac{1}{s+1} & 0 & 0 \\ 0 & 0 & \frac{1}{s+1} & 0 \\ 0 & 0 & 0 & \frac{1}{s+1} \end{bmatrix} \quad (6.29)$$

suivant le même raisonnement que pour le cas longitudinal. L'ordre du correcteur a été fixé à 2 et la résolution du problème de minimisation \mathcal{H}_∞ (4.14)-(4.15) conduit à une norme \mathcal{H}_∞ optimale $\gamma_{opt} = 1.03$.

Une LFT du système est ensuite déterminée, permettant d'évaluer la stabilité robuste de l'ensemble avec ce correcteur nominal $\hat{K}_0(s)$. La taille du bloc d'incertitudes Δ correspondant est de 24×24 , et la μ -analyse donne dans un premier temps une valeur singulière structurée comprise entre $\underline{\mu}_\Delta = 1.007$ et $\bar{\mu}_\Delta = 1.118$, amenant la conclusion que toutes les configurations ne sont pas stabili-

sées. L'algorithme 5 est alors exécuté, permettant d'aboutir à un nouveau correcteur multi-modèles $\hat{K}(s)$ qui, après 3 itérations, donne la valeur singulière structurée $\mu_{\Delta} \in [0.7948 \ 0.8845] < 1$ assurant la stabilité robuste quelque soit la configuration et malgré les incertitudes de modélisation. Il peut être intéressant de remarquer que lors de ces itérations, la borne supérieure ne s'est jamais révélée trop conservatrice et aucun algorithme de branch-&-bound n'a été employé.

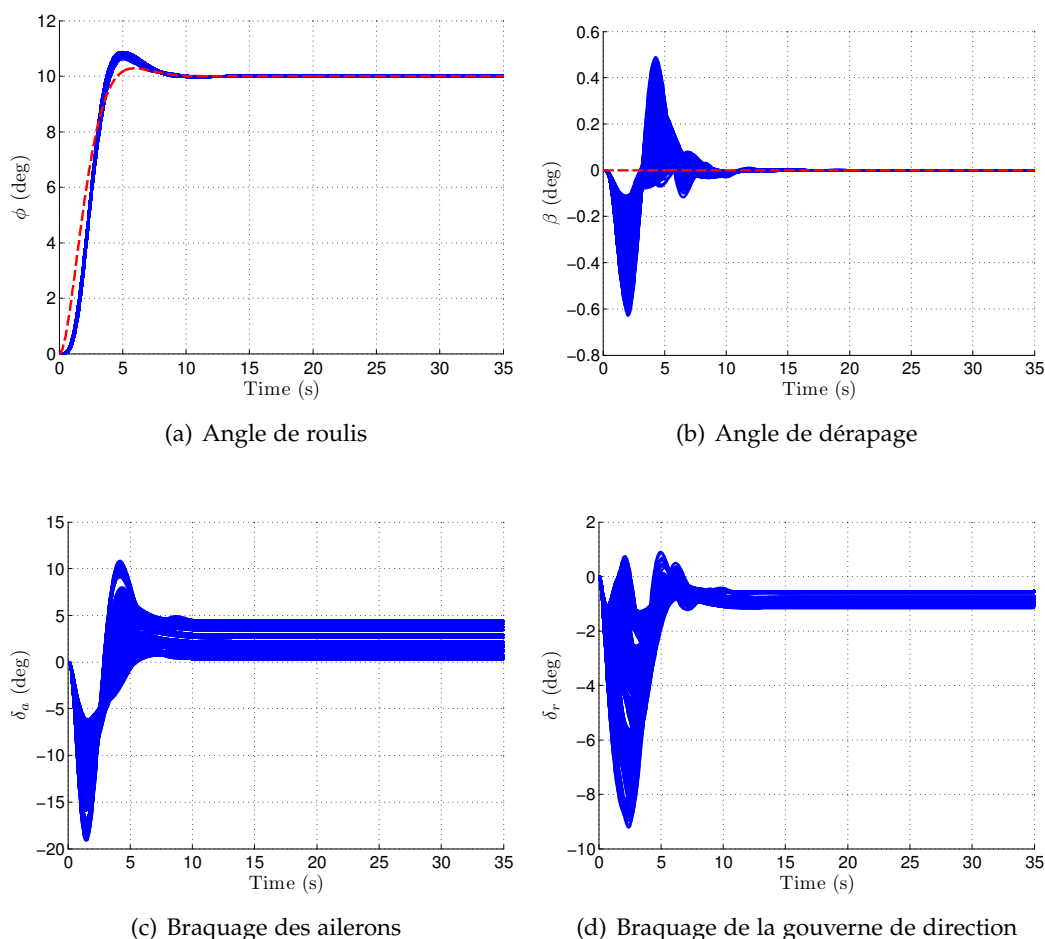


FIGURE 6.6 – Simulations sur le modèle non-linéaire complet, avec un échelon de 10 degrés sur l'angle de roulis ϕ , pour différentes configurations paramétriques du domaine admissible Θ , et $\pm 30\%$ d'incertitudes sur les coefficients aérodynamiques latéraux, avec le correcteur multi-modèles $\hat{K}(s)$.

Des simulations avec le correcteur $\hat{K}(s)$ sont lancées pour différentes configurations appartenant au domaine admissible Θ et avec $\pm 30\%$ d'incertitudes sur les coefficients aérodynamiques latéraux, amenant 729 exécutions au total. Elles sont présentées aux Figures 6.6 et 6.7, où des sollicitations respectivement en roulis ϕ et en dérapage β sont très bien suivies grâce à la loi de commande développée malgré les diverses configurations et incertitudes, avec des entrées de commande raisonnable. Le découplage est également très bien assuré, et aucune instabilité ou

début d'instabilité n'est à signaler. Ceci confirme également l'efficacité de la méthode pour le pilotage latéral robuste d'un avion civil.

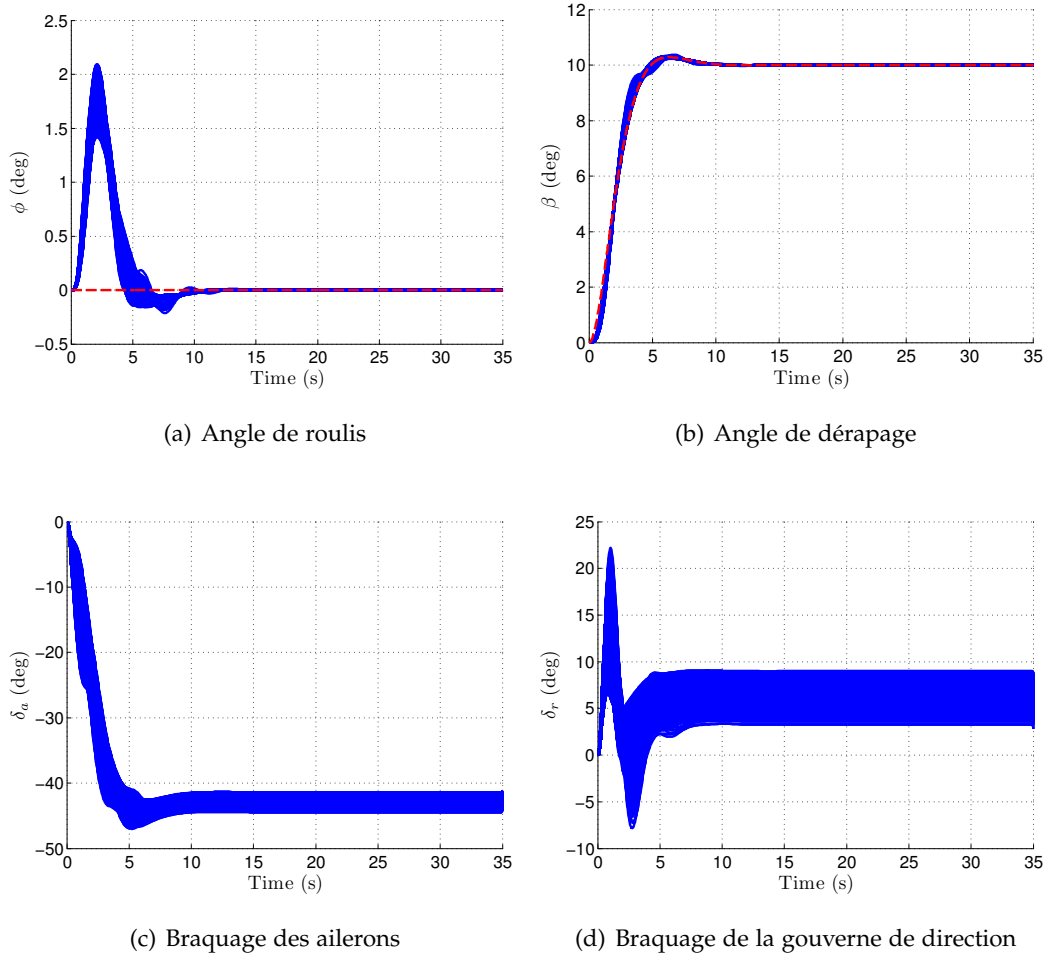


FIGURE 6.7 – Simulations sur le modèle non-linéaire complet, avec un échelon de 10 degrés sur l'angle de dérapage β , pour différentes configurations paramétriques du domaine admissible Θ , et $\pm 30\%$ d'incertitudes sur les coefficients aérodynamiques latéraux, avec le correcteur multi-modèles $\hat{K}(s)$.

6.3 GUIDAGE POUR L'ATTERRISSAGE AUTOMATIQUE PAR SUIVI D'ILS

Après avoir synthétisé les lois de commande pour le modèle réaliste d'un avion civil, cette section vise à détailler la loi de guidage en vue de l'atterrissage de ce dernier. Cette phase de vol d'un avion est considérée comme la plus critique, notamment les parties finales d'arrondi et de décroche. Il existe cependant des systèmes d'atterrissage automatique aux instruments, ou ILS, notamment ceux de Catégorie III qui sont présents sur de nombreux aéroports, qui permettent d'atterrir de façon automatique même avec des conditions météorologiques compliquées (visibilité réduite, présence de vent latéral, ...). Le principe général du guidage par ILS

sera brièvement expliqué, et seront surtout développées les lois de guidage pour le suivi des signaux ILS et pour les phases d'arrondi et de décroche. Un lot de simulations viendra tester l'ensemble guidage + commande malgré les différentes configurations de vol, la présence d'incertitudes de modélisation, et de vent.

6.3.1 ILS et guidage

Principe général et objectifs

L'ILS est composé d'émetteurs, de petites stations sol, et d'équipement à bord de l'avion permettant la réception des signaux. Le système sol, situé à l'entrée de piste, est constitué d'émetteurs radio permettant de transmettre les signaux dits "glide" et "localizer" à l'avion, ainsi que de marqueurs lumineux. A bord de l'avion, les récepteurs permettent de traiter les signaux radio pour notamment en déduire les écarts longitudinaux et latéraux par rapport à la trajectoire idéale à suivre en vue de l'atterrissage. Ainsi les signaux glide et localizer sont ceux qui nous intéressent en vue de connaître les écarts relatifs respectivement longitudinaux et latéraux qu'il va falloir chercher à corriger, par une boucle de guidage. Celle-ci doit assurer que l'avion suive une trajectoire rectiligne centrée sur la piste et avec une pente de descente de -3 degrés. Les principes sont présentés schématiquement sur les figures 6.8 et 6.9. On y voit notamment l'écart vertical dZ et l'écart latéral dY par rapport à la trajectoire de référence centrée sur la piste de pente -3 degrés. Plus de détails à ce sujet peuvent être trouvés notamment dans [Looye et Joos 2006].

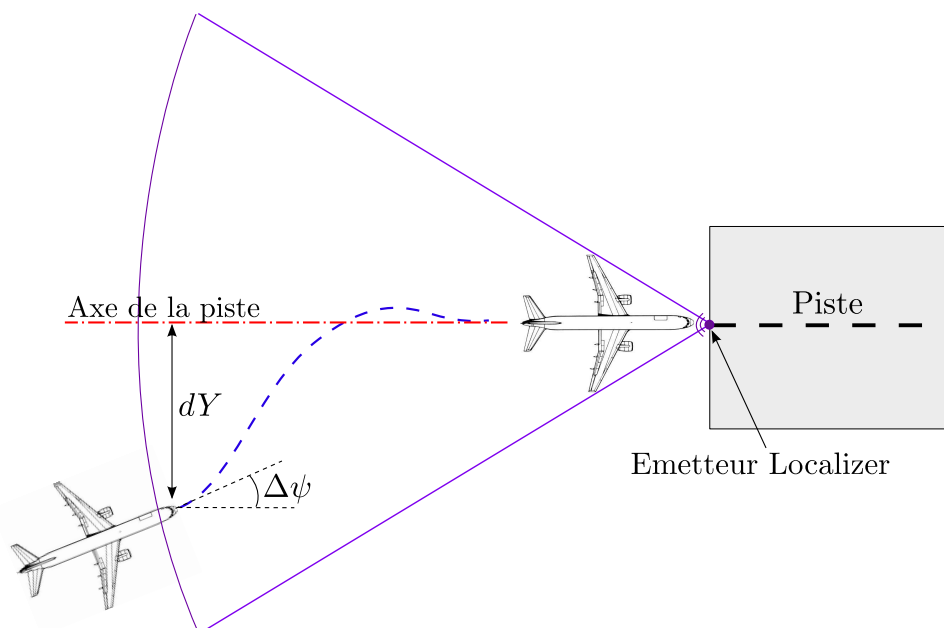


FIGURE 6.8 – Signal "localizer" en vue de la correction de la déviation latérale de l'avion par rapport à l'axe de la piste d'atterrissage.

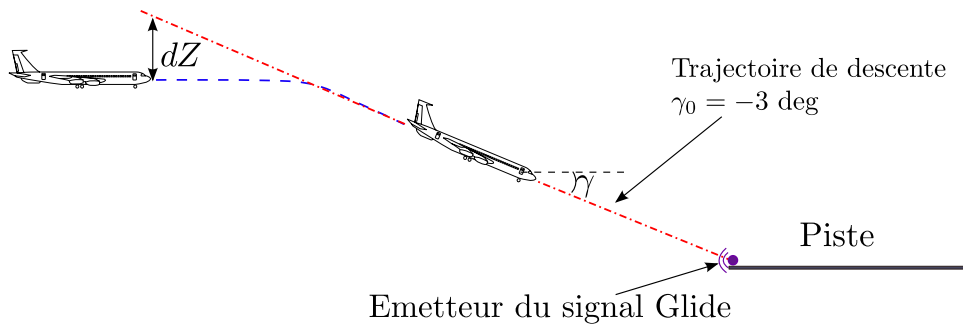


FIGURE 6.9 – Signal "glide" en vue de la correction verticale de l'avion par rapport à la pente optimale de descente pour l'atterrissage.

Le but final est de permettre à l'avion de toucher idéalement le sol avec une vitesse verticale V_z comprise entre 2 et 3 pieds par seconde (entre 0.6 et 0.9 m/s environ), et à une distance inférieure à 500 mètres du seuil de piste. De plus, en présence de vent latéral, le cap de l'avion doit être ramené dans l'axe de la piste (on notera ψ ce cap et $\Delta\psi$ l'écart par rapport à l'alignement de la piste) sans que l'avion possède un angle de roulis trop fort et se dévie par rapport à l'axe de piste. Ces objectifs doivent également être vérifiés malgré les différentes configurations de vol et incertitudes.

Lois de guidage

Afin de corriger les éventuels écarts dZ et dY , deux correcteurs PID sont calculés. Ils permettent de générer respectivement des consignes de pente γ_c et de roulis ϕ_c afin de maintenir $dZ = dY = 0$. Pour ce faire des modèles linéarisés sont obtenus à partir du modèle complet, entre dZ et γ_c d'une part, et entre dY et ϕ_c d'autre part, autour du point de fonctionnement nominal $\bar{\theta}_p$ utilisé jusqu'alors.

La vitesse air est maintenue constante, à $V_c = 70$ m/s, et l'angle de dérapage est également fixé à $\beta_c = 0$ degrés afin de garantir un vol symétrique en approche.

Remarque 6.5

- Quelques remarques méritent d'être mentionnées à propos de ces stratégies de guidage.
- pour le guidage vertical (correction de dZ), si l'avion est en dessous de la trajectoire optimale (i.e. $dZ > 0$ vu que l'axe vertical est dirigé vers le bas), il serait peu judicieux de forcer l'avion à remonter pour rattraper la pente de descente à -3 degrés. En effet ceci amènerait une consommation de carburant dispensable, alors qu'il suffit d'imposer une trajectoire à pente nulle ($\gamma_c = 0$) voire de pente inférieure, et d'attendre que l'avion coupe la trajectoire pour activer la correction (comme représenté à la figure 6.9),
 - dans le cas où l'écart initial dZ est élevé, et que l'avion est encore assez loin de la piste, il n'est pas non plus nécessaire de corriger rapidement la trajectoire, d'autant plus que les gains du PID sont forts pour maintenir une certaine précision

afin que la phase d'arrondi soit bien réalisée. Ainsi une saturation est ajoutée sur l'écart dZ pour que la correction reste douce tout en ne limitant pas les gains du PID lorsque l'avion arrive près de la piste,

- enfin pour le guidage latéral, en présence de vent de travers, étant donné qu'est imposé $\beta = 0^\circ$, l'avion aura tendance à se mettre en "crabe". Ceci implique que le cap de l'avion ψ n'est plus aligné avec l'axe de piste ($\Delta\psi \neq 0^\circ$), même si sa trajectoire suit correctement ce dernier. Ceci est bien évidemment à corriger au moment de l'arrondi, lors de la phase dite de "décrabe", pour que l'avion atterrisse bien dans l'axe (voir prochaine sous-section).

6.3.2 Phases d'arrondi et de décrabe

Juste avant l'atterrissage final, l'avion est normalement correctement aligné et sur sa trajectoire, grâce aux lois de guidage précédemment développées : vitesse air de 70 m/s, angle de vol à -3 degrés, et centré sur la piste. Il reste maintenant à assurer les phases d'arrondi et d'alignement avec la piste (ou décrabe), dans le cas où il persiste un vent latéral. La figure 6.10 présente l'ensemble des blocs de guidage et de commande, comprenant aussi le bloc d'arrondi et de décrabe.

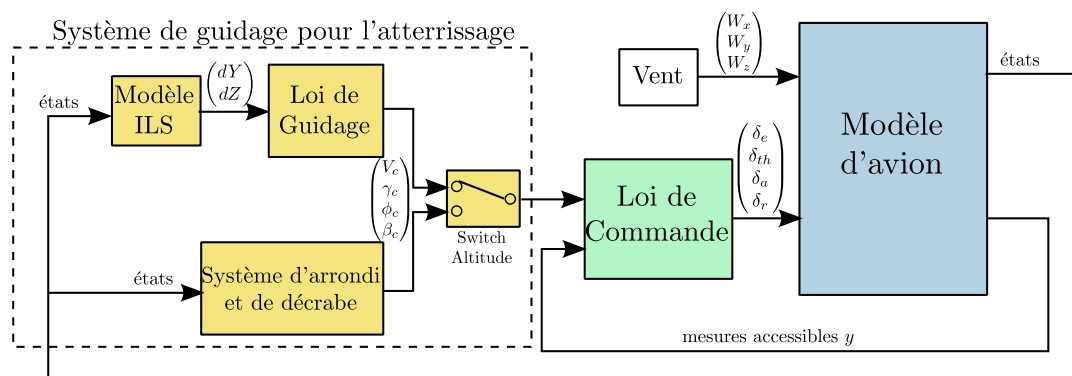


FIGURE 6.10 – Schéma global de commande et de guidage pour l'atterrissage automatique.

Arrondi

L'arrondi est généralement exécuté à partir d'une altitude d'environ 50 pieds (15 mètres) au-dessus du seuil de piste. Il consiste à générer une consigne γ_c qui permette à l'avion de suivre une trajectoire telle qu'au toucher des roues, il ait une vitesse verticale $V_z = 2.5$ pieds/secondes (environ 0.75 m/s), tout en assurant qu'il touche le sol entre 300 et 500 mètres après le seuil de piste. Pour ce faire, la stratégie consiste à utiliser la relation suivante, exprimant la vitesse verticale V_z en fonction de l'angle de vol γ :

$$V_z = V_a \sin \gamma \quad (6.30)$$

où la vitesse air V_a est maintenue constante par la loi de commande longitudinale. Il est ensuite possible de déduire la valeur finale γ_c que doit atteindre l'avion pour assurer une vitesse verticale donnée. Le profil en angle de vol à suivre est un premier ordre de constante de temps τ_γ qui sera dépendante de la vitesse sol mesurée V_{gr} telle que :

$$\tau_\gamma = \frac{x_l}{3 * V_{gr}} \quad (6.31)$$

où x_l est la distance souhaitée de toucher des roues de l'avion (typiquement 450m à partir du seuil de piste). Cette dépendance est particulièrement utile s'il existe un vent longitudinal de face plus ou moins fort. On peut aussi évaluer la distance au point d'impact souhaité lorsqu'on arrive à l'altitude où doit être déclenché l'arrondi, et ainsi adaptée x_l en fonction pour avoir un profil de γ_c plus approprié (dans le cas où l'avion est légèrement décalé à cause de perturbations par rapport au signal glide de l'ILS). Un switch permet de passer du suivi du signal glide à la phase d'arrondi, dès que l'altitude devient inférieure à 50 pieds.

Décrabe

En présence de vent latéral, au cours de la phase d'approche, le pilote cherche à maintenir un angle de dérapage β nul, ce qui risque d'amener un cap ψ différent du cap d'alignement de la piste, dont l'écart est noté $\Delta\psi$. Afin de ne pas entraîner d'incident lors du toucher des roues au moment de la phase d'arrondi, il faut remettre l'avion dans l'axe à partir d'environ 30 pieds d'altitude au dessus de la piste.

Pour ce faire, un simple correcteur PI est dimensionné, ayant $\Delta\psi$ en entrée, et sortant l'angle de dérapage correspondant $\beta_c \neq 0$ qui est cette fois envoyé à la loi de commande latéral. On passe donc, grâce à une commutation qui s'active lorsque l'altitude de l'avion devient inférieure à 30 pieds, de $\beta_c = 0$ à :

$$\beta_c = k_p^d \Delta\psi + k_i^d \int \Delta\psi \quad (6.32)$$

afin d'obtenir $\Delta\psi = 0$. Le réglage des gains k_p^d et k_i^d devra permettre une réponse suffisamment rapide mais pas trop brutale, afin que l'avion n'ait pas un angle de gauchissement trop important ni une déviation par rapport à l'axe de piste trop conséquente.

6.3.3 Simulations temporelles

Cette sous-section propose de valider l'approche par simulations temporelles. Dans un premier temps, des simulations d'atterrissage complet sont présentées sur la figure 6.11, pour quelques configurations de l'avion avec des incertitudes. Ce

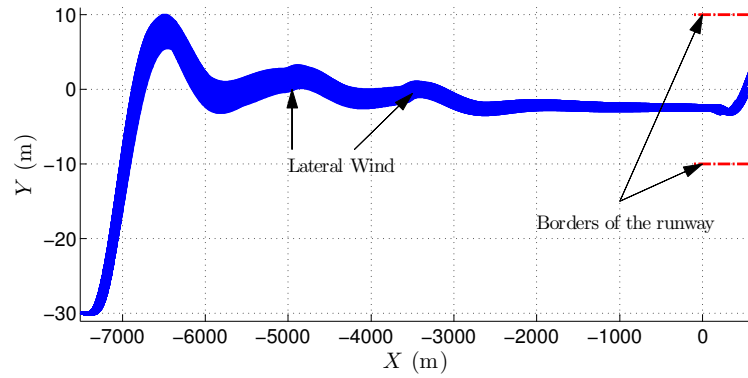
dernier est initialement décalé longitudinalement et latéralement de la pente idéale d'approche et de l'alignement de la piste, d'une valeur de 30 mètres sur chaque axe. Le système de guidage le ramène donc dans l'axe avec un angle de vol adapté (-3 degrés) (comme le montrent les sous-figures 6.11(a), 6.11(b) et 6.11(e)). Il est à noter la présence de deux rafales de vent latéral apparaissant à 35 secondes et 55 secondes, induisant un vol "en crabe" afin de garder l'angle de dérapage nul. Les phases d'arrondi et de décroche sont réalisées une fois l'avion au-dessus du seuil de piste, permettant à ce dernier d'atterrir dans de bonnes conditions en ce qui concerne notamment la vitesse verticale au toucher des roues (commandée via l'angle de vol γ , sous-figure 6.11(e)), l'angle de cap ramené à celui de la piste (voir sous-figure 6.11(d)) ou encore la distance d'atterrissage (voir zoom présentant l'arrondi sur la sous-figure 6.11(b)). Il faut également veiller à ce que l'angle de roulis ne soit pas trop élevé au moment du toucher des roues, de façon à ce que les moteurs ne percutent pas le sol (voir sous-figure 6.11(c)).

Après avoir illustré pour quelques configurations l'approche complète, il est proposé dans un deuxième temps de réaliser des simulations de Monte Carlo afin de valider statistiquement la méthode. Les paramètres variant aléatoirement concernent :

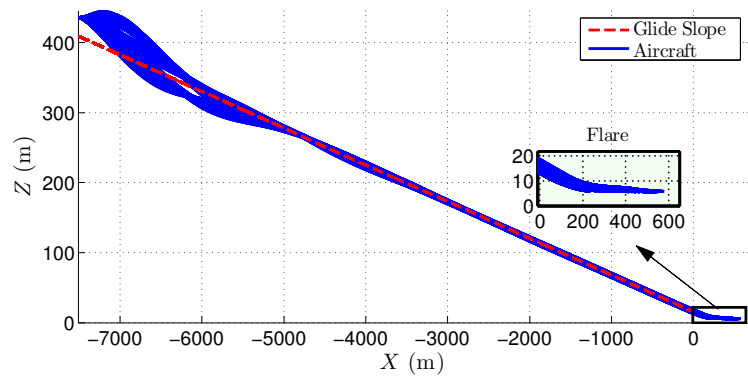
- la configuration initiale de l'avion (masse et centrage),
- les incertitudes sur les coefficients aérodynamiques,
- les différents profils de vent, constitués à la fois de turbulences sur tous les axes dont l'amplitude est choisie aléatoirement pour un profil gaussien variant, et également de rafales apparaissant à 150 mètres d'altitude pour le latéral (mettant l'avion en crabe), à 15 mètres pour le longitudinal, et à 5 mètres pour le vertical (perturbant ainsi l'arrondi).

10000 simulations sont lancées, et une étude probabiliste sur 6 paramètres de sortie, évalués lors du toucher des roues, est réalisée. Ces paramètres sont :

- *HTP60* : hauteur du train d'atterrissage 60 mètres avant le seuil de piste, qui doit être suffisamment élevée pour ne pas risquer un atterrissage prématuré,
- *XTP* : distance d'atterrissage (coordonnée du point d'impact des roues selon l'axe x), qui ne doit pas dépasser une certaine limite au risque de ne pas pouvoir freiner sans sortir de piste,
- *VZTP* : vitesse verticale à l'impact, qui ne doit être ni trop faible pour assurer un bon toucher des roues, ni trop forte pour ne pas impacter sévèrement le train et la structure de l'avion,
- *YTP* : déviation latérale de l'avion (due au vent transverse), qui ne doit pas être trop élevée pour rester dans les limites de la piste,



(a) Trajectoire latérale



(b) Trajectoire longitudinale

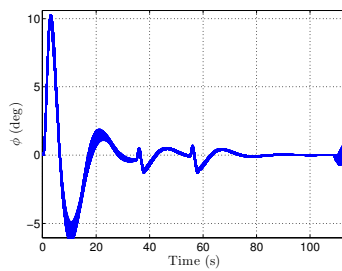
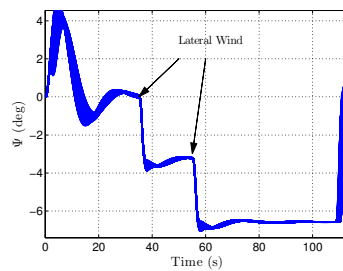
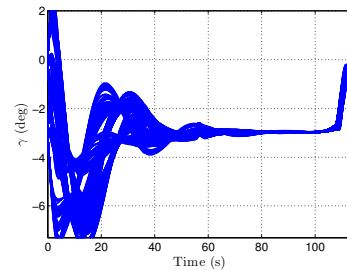
(c) Variations de ϕ (d) Variations de ψ (e) Variations de γ

FIGURE 6.11 – Simulations d'atterrissage complet en présence de vent latéral, pour différentes configurations de l'avion.

- *PHI* : angle de roulis au moment du toucher des roues, qui ne doit pas dépasser une certaine valeur afin que les moteurs ne touchent pas le sol,
- *SSTP* : angle de dérapage du train, qui ne doit pas être trop élevé.

Les spécifications de tous ces paramètres sont données au tableau 6.2, avec notamment la valeur demandée (la valeur moyenne doit idéalement être contenue dans cette plage), et la probabilité $\mathcal{P}(\mathcal{A})$ qu'une condition \mathcal{A} ait une chance inférieure à 10^{-6} de se produire, assurant qu'il y a à priori un risque statistiquement quasi inexistant de dépasser certaines valeurs critiques.

Exigences	Paramètre	Demande	Risques à vérifier
Longitudinales	<i>HTP60</i> (m)	> 0	$\mathcal{P}(HTP60 < 0) < 10^{-6}$
	<i>XTP</i> (m)	$[300 \ 500]$	$\mathcal{P}(XTP > 915) < 10^{-6}$
	<i>VZTP</i> (ft/s)	$[2 \ 6]$	$\mathcal{P}(VZTP > 10) < 10^{-6}$
Latérales	<i>YTP</i> (m)	$[0 \ 5]$	$\mathcal{P}(YTP > 10) < 10^{-6}$
	<i>PHI</i> (deg)	$[-1 \ 1]$	$\mathcal{P}(PHI > 8) < 10^{-6}$
	<i>SSTP</i> (deg)	$[-1 \ 1]$	$\mathcal{P}(SSTP > 10) < 10^{-6}$

TABLEAU 6.2 – Spécifications à tenir pour l'atterrissage, lors du toucher des roues.

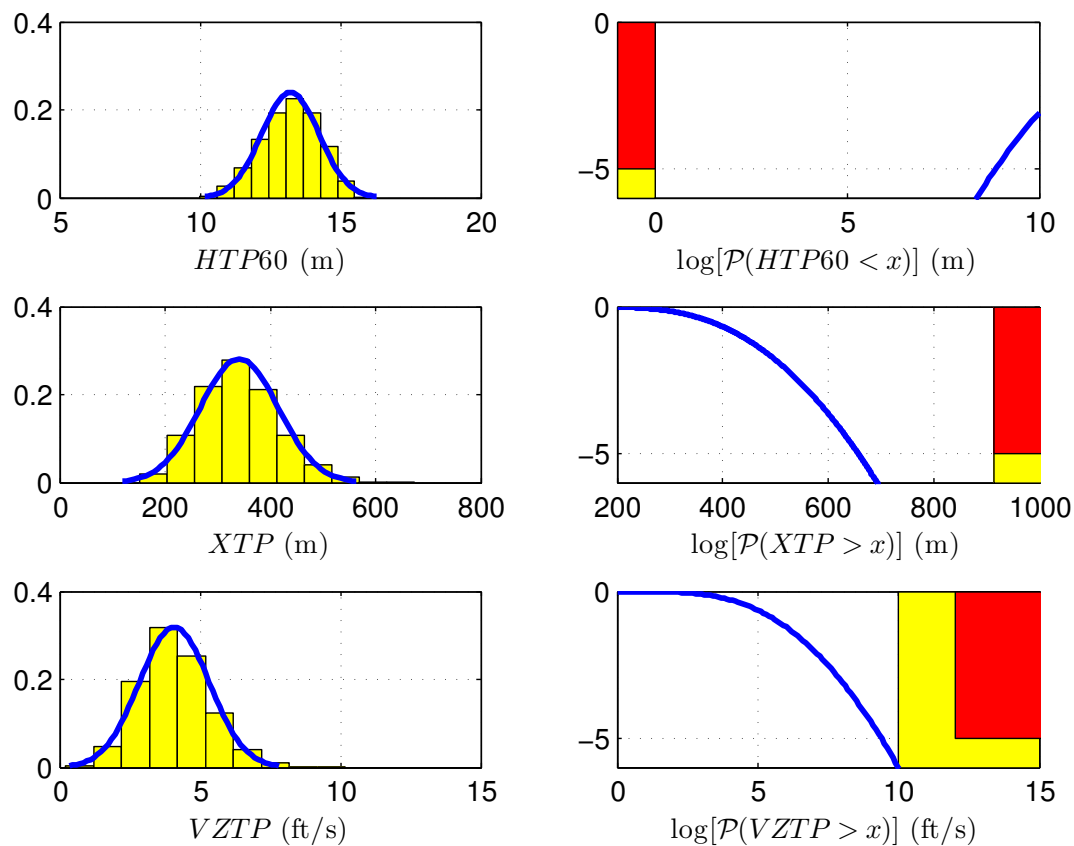


FIGURE 6.12 – Interpolations gaussiennes et probabilités qu'une valeur atteigne le seuil critique après 10000 simulations de Monte Carlo : exigences longitudinales.

Les résultats ¹ sont présentés sur les figures 6.12 et 6.13, correspondant respectivement aux exigences longitudinales et latérales. Sur chaque figure sont disposées à gauche les interpolations gaussiennes avec les répartitions sous forme d'histogrammes de chaque variable, permettant d'évaluer la valeur moyenne ainsi que l'écart type, et à droite les courbes $\log(\mathcal{P}(\mathcal{A}(x)))$, où x est l'abscisse, avec la zone colorée en rouge à ne pas atteindre (correspondant à un risque supérieur à 10^{-5} de contrarier les exigences imposées au tableau 6.2), et la zone jaune à éviter le plus possible (correspondant à un risque compris entre 10^{-5} et 10^{-6}). Par exemple, on peut lire sur la figure 6.12 que la probabilité d'atterrir à plus de 650 mètres du seuil de piste est de 10^{-5} , et qu'à priori on atterrira plus loin que 700 mètres avec un risque inférieur à 10^{-6} , ce qui permet de valider ce critère. On observe ainsi globalement que toutes les exigences sont tenues, avec cependant un risque très limite concernant la vitesse verticale. Cette étude permet donc de valider l'approche malgré la présence de vents (turbulences et rafales), de changements de configuration de l'appareil, et des incertitudes de modélisation.

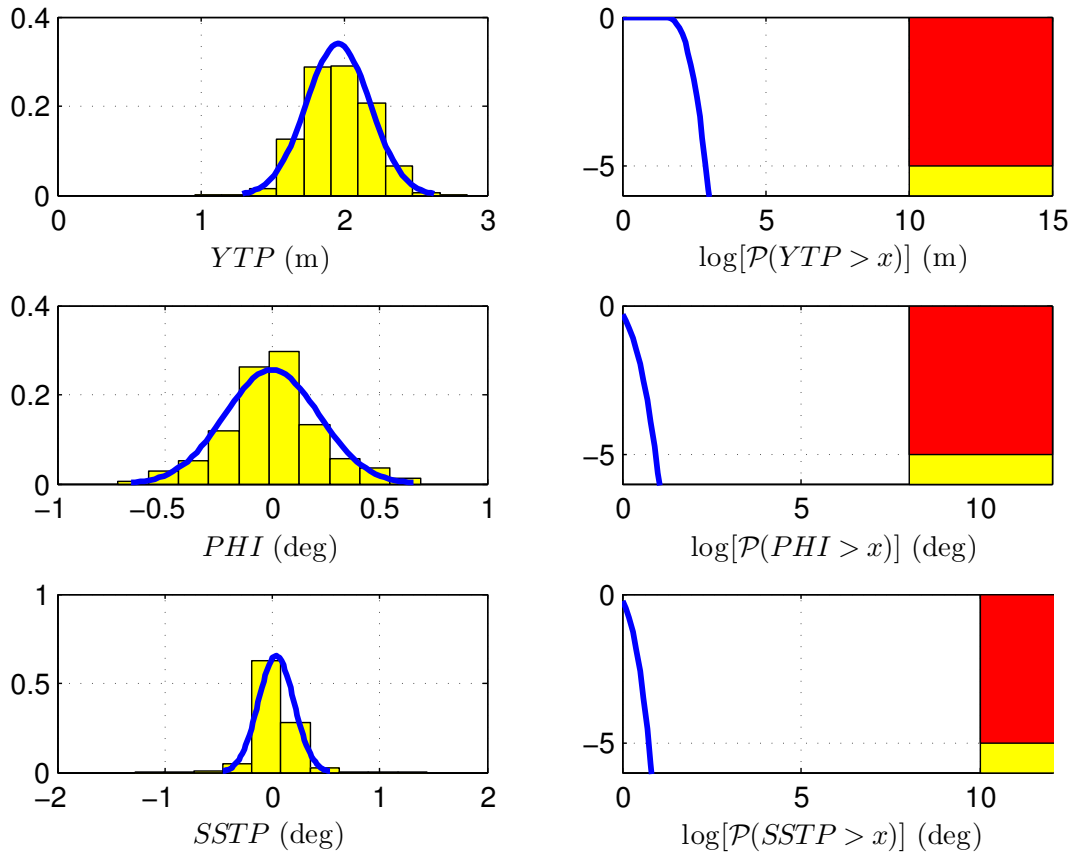


FIGURE 6.13 – Interpolations gaussiennes et probabilités qu'une valeur atteigne le seuil critique après 10000 simulations de Monte Carlo : exigences latérales.

1. Les outils permettant d'obtenir ces courbes de Monte Carlo sont disponibles sur le site de la boîte à outils SMAC [SMAC 2012-2015].

Remarque 6.6

Une façon d'améliorer sensiblement les résultats – notamment concernant la vitesse verticale – serait de dimensionner une loi de commande robuste sur le modèle linéarisé près du sol, ayant donc l'effet de sol pris en considération. Ainsi, un switch permettrait de passer de la loi de pilotage en croisière à celle au moment de l'atterrissage, dès lors que l'avion se situe suffisamment près du sol. Néanmoins, durant ces travaux, il était intéressant de montrer qu'une seule loi de pilotage permettait de traiter avec succès un domaine de fonctionnement assez large.

CONCLUSION DU CHAPITRE

La méthode de compensation non-linéaire robuste développée au chapitre 4, faisant également appel aux techniques développées dans la partie 1 de ce manuscrit, a été appliquée dans le présent chapitre à un problème de pilotage automatique d'un modèle d'avion civil complet. La prise en compte des pire-cas lors de la synthèse \mathcal{H}_∞ permet d'améliorer sensiblement la robustesse de la loi globale, et la boucle interne linéarisante adapte très bien la commande à de nombreux points de fonctionnement. Elle est donc parfaitement adaptée pour le pilotage robuste d'un engin aéronautique sur l'ensemble de son domaine de vol.

De plus, une loi de guidage a également été développée afin de proposer une application complète d'atterrissage automatique, allant du suivi d'ILS au toucher des roues, en passant par la phase d'arrondi et de décroche en présence de vent. La méthode s'avère particulièrement efficace pour répondre aux exigences de fiabilité et de sûreté dans cette phase de vol pourtant très critique.

MODÉLISATION ET PILOTAGE ROBUSTE D'UN DRONE À VOILURE FIXE

SOMMAIRE

7.1	DESCRIPTION DU DRONE "AVION JAUNE"	159
7.2	MODÉLISATION NUMÉRIQUE	160
7.2.1	Détermination des paramètres structuraux	160
7.2.2	Détermination des coefficients aérodynamiques	161
7.2.3	Modélisation des élévons	163
7.2.4	Modélisation du bloc de propulsion	164
7.2.5	Simulations en boucle ouverte et comparaison avec des données de vol	167
7.3	SYNTHÈSE DE LOIS DE PILOTAGE AUTOMATIQUE	173
7.3.1	Synthèse modale	173
7.3.2	Technique de compensation non-linéaire robuste	174
7.3.3	Simulations temporelles	176
7.4	DE L'IMPLANTATION RÉELLE DES LOIS	177
	CONCLUSION	178

Résumé

UNE procédure générique de modélisation d'un drone à voilure fixe ainsi que l'application de la méthode de synthèse non-linéaire robuste développée au chapitre 4 sont proposées dans ce chapitre. Un modèle numérique est facilement obtenu en se basant sur les équations de la mécanique du vol présentées au chapitre 5, comprenant également la modélisation du bloc de motorisation et des actionneurs. Il est validé par comparaison avec des essais en vol. Ceci permet par la suite de dimensionner une loi de commande pour le pilotage longitudinal du drone, la méthode de synthèse proposée dans ce mémoire se révélant particulièrement pertinente et de plus facilement implantable.

Les drones à voilure fixe sont utilisés dans beaucoup de domaines, qu'ils soient militaires ou civils. On peut notamment citer des opérations de recherche et de sauvetage, d'exploration et de détection, de la surveillance, ou encore de l'imagerie et de la photographie. La plupart de ces opérations ont lieu dans des endroits souvent inaccessibles voire dangereux.

Le succès de telles opérations dépend, entre autres, des systèmes de commande qui doivent s'avérer à la fois robustes et performants. Les lois de pilotage synthétisées doivent faire face au comportement fortement non-linéaire du drone, agir efficacement sur l'ensemble du domaine d'opération qui peut être large, rejeter correctement les perturbations atmosphériques, et être robuste aux éventuelles incertitudes de modélisation.

La méthode de synthèse développée au chapitre 4 peut s'avérer efficace et intéressante pour répondre à ces objectifs. Elle a prouvé son efficacité pour le pilotage d'un avion civil en phase d'atterrissage (chapitre 6) : est-il possible d'adapter la méthode facilement pour le pilotage d'un drone à voilure fixe ? C'est ce que vise à présenter ce chapitre.

Il est organisé comme suit. Dans un premier temps, le drone à voilure fixe utilisé est présenté à la section 7.1. Afin de pouvoir dimensionner des lois de pilotage, il est important dans un premier temps d'obtenir une modélisation fiable et représentative de l'engin. La section 7.2 se concentre sur ce point, en se basant notamment sur les équations de la mécanique du vol générale présentées au chapitre 5. Des comparaisons sont également exposées entre le modèle simulé et des essais en vol en boucle ouverte pour valider le modèle. Ensuite une technique de synthèse modale (placement de structure propre) d'une part, et la technique de synthèse non-linéaire robuste du chapitre 4 d'autre part, sont appliquées pour synthétiser un correcteur longitudinal à la section 7.3. Enfin quelques éléments sont donnés à la section 7.4 quant à la possibilité d'implanter le correcteur non-linéaire robuste obtenu sur le système réel afin de réaliser des essais en vol en boucle fermée.

7.1 DESCRIPTION DU DRONE "AVION JAUNE"

L'Avion Jaune Altimum 4¹, disponible à l'ONERA, est un drone à voilure fixe à motorisation thermique arrière qui peut emporter jusqu'à 4kg de charge utile et opérer au sein d'un large rayon d'action (100km). Il a été conçu pour pouvoir réaliser des prises de vue et des mesures atmosphériques au niveau de zones marines, et peuvent prendre part dans des conditions météorologiques perturbées. Il a la forme d'une aile volante de 4 mètres d'envergure, voir figure 7.1. La plage de

1. http://www.lavionjaune.fr/fr/3b_drones.html



FIGURE 7.1 – Drone aérien à voilure fixe, l'Avion Jaune.

vitesse considérée va de 18 m/s à 28 m/s, celle de masse de 17 kg à 23 kg, et le centre de gravité peut varier de $\pm 5\%$.

Étant donné qu'il s'agit d'une aile volante sans stabilisateurs, le drone dispose d'élevons pour gérer à la fois les ailerons (roulis, braquage δ_a) et la profondeur (tangage, braquage δ_e). Afin de se ramener aux notations standards de la mécanique du vol, les braquages des élevons gauche et droit, respectivement notés δ_l et δ_r , sont liés à δ_a et δ_e par :

$$\delta_l = \delta_a - \delta_e \quad (7.1)$$

$$\delta_r = -\delta_a - \delta_e \quad (7.2)$$

De plus, le drone ne dispose pas de dérive, donc il n'y a pas de commande en δ_r . Des dérives seront cependant ajoutées prochainement en bout d'aile au niveau des winglets.

Remarque 7.1

Les données concernant le profil d'aile ainsi que les coefficients aérodynamiques évalués dans ce chapitre étant confidentiels, aucune information quant à ceux-ci n'est fournie dans ce rapport.

7.2 MODÉLISATION NUMÉRIQUE

7.2.1 Détermination des paramètres structuraux

Dans un premier temps il convient de déterminer la matrice d'inertie I et la masse m (présentes dans l'équation de la dynamique (5.18)). Cette dernière est facilement mesurable en pesant tous les composants de l'avion un à un. La matrice d'inertie est quant à elle déterminée de façon numérique grâce à une modélisation sur un outil de conception assistée par ordinateur comme CATIA (voir figure 7.2). Il

faut bien cependant déterminer les masses et positions des autres composants, tels que les servo-commandes, le moteur, les batteries, les récepteurs, et autres charges utiles, afin que le logiciel les prenne en compte pour le calcul des inerties.

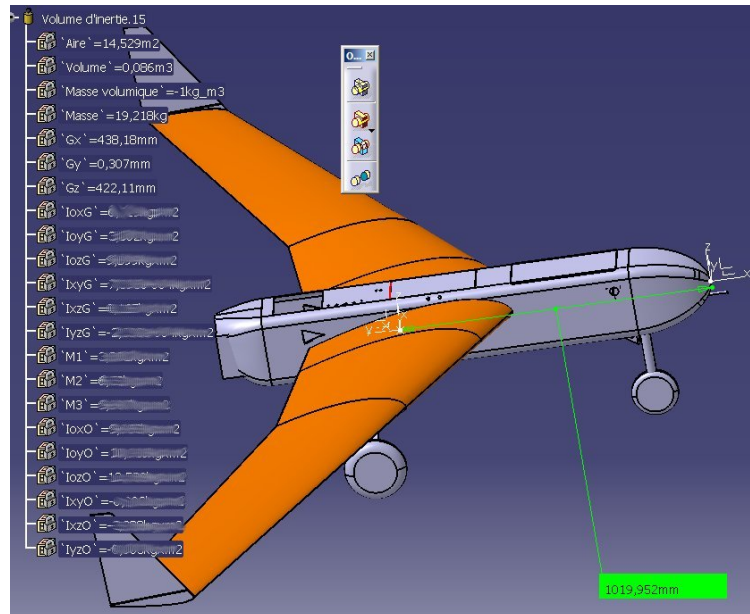


FIGURE 7.2 – Modélisation de l'Avion Jaune sur CATIA, en vue de l'obtention numérique de la matrice d'inertie (les valeurs étant cachées par souci de confidentialité).

Enfin il est nécessaire pour l'expression des moments (sous-section 5.3.2) d'avoir la position du centre de gravité ainsi que la position relative du moteur par rapport au centre de gravité (constantes x_{GE} et z_{GE}). La position du centre de gravité est mesurée sur l'avion de façon expérimentale par un système de balance une fois l'avion suspendu. Elle est déterminée de façon relativement précise. Il est également possible de la comparer à l'évaluation numérique faite sous CATIA (voir figure 7.2), la valeur étant dans notre cas très proche de la valeur expérimentale (ce qui est rassurant!). La position du moteur est quant à elle facilement mesurée, en notant bien ici que $z_{GE} < 0$ du fait que le moteur est situé "au-dessus" du centre de gravité, ce qui va avoir tendance à entraîner un moment à piquer dû à la poussée moteur (voir remarque 5.3).

7.2.2 Détermination des coefficients aérodynamiques

Les équations des forces et des moments font apparaître un certain nombre de coefficients aérodynamiques qu'il faut évaluer (voir section 5.5). On négligera l'effet de sol, impossible à numériser précisément numériquement, ce qui de toute façon n'a que peu d'importance compte tenu qu'on ne s'intéresse pas ici à la phase d'atterrissage.

Les coefficients C_{L0} , $C_{L\alpha}$, C_{Lq} , $C_{L\delta_e}$, $C_{Y\beta}$, C_{Yp} , C_{Yr} , $C_{Y\delta_r}$, $C_{Y\delta_a}$, C_{D0} , C_{DC_L} , $C_{l\beta}$, C_{lp} , C_{lr} , $C_{l\delta_a}$, C_{m0} , $C_{m\alpha}$, C_{mq} , $C_{m\delta_e}$, $C_{n\beta}$, C_{nr} , C_{np} et $C_{n\delta_a}$ des équations (5.37)-(5.42) peuvent

être dans un premier temps évalués en utilisant une soufflerie numérique comme XFLR5, qui fonctionne à faible nombre de Reynolds (ce qui est une hypothèse réaliste compte tenue du domaine de fonctionnement du drone). Il faut bien entendu noter que, l'avion ne disposant pour le moment pas de commande de direction δ_r , les coefficients $C_{Y_{\delta_r}}$, $C_{l_{\delta_r}}$ et $C_{n_{\delta_r}}$ sont nuls. L'utilisation de XFLR5 requiert une modélisation 3D de l'appareil, qui est facilement réalisée grâce aux données géométriques fournies par l'industriel. De plus, les différents profils aérodynamiques sont disponibles dans des bases de données assez complètes², permettant de modéliser l'aile avec précision. Le modèle obtenu est présenté à la figure 7.3.

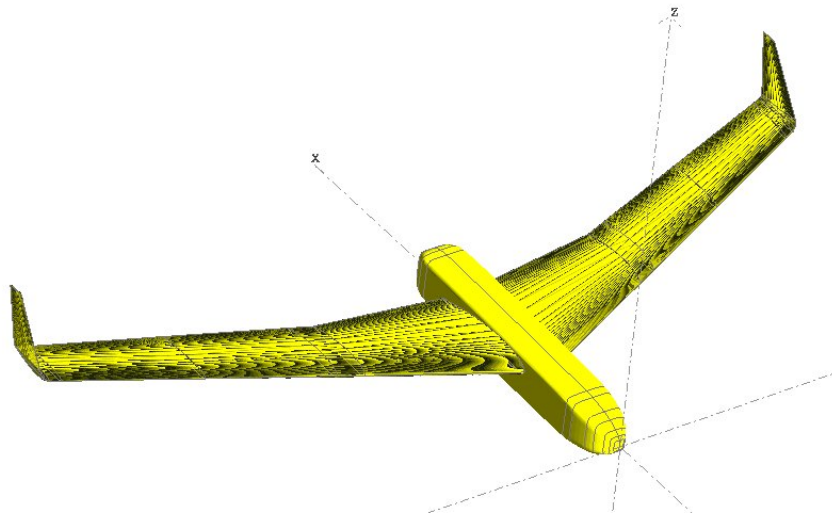


FIGURE 7.3 – Représentation 3D réalisée sous XFLR5 du drone à voilure fixe.

Une fois la modélisation 3D réalisée, XFLR5 permet d'une part d'approximer la matrice I , et d'autre part d'utiliser différents algorithmes (adaptés en fonction des cas) afin de déterminer les coefficients aérodynamiques. On peut citer par exemple :

- la théorie de la ligne portante (LLT), qui permet d'exprimer la distribution de portance d'une aile comme une fonction non-linéaire de l'angle d'attaque α [Sivells et Neely 1947]. L'hypothèse principale de cette méthode est de remplacer l'aile par une ligne portante générant des tourbillons qui changent la valeur effective de l'angle d'attaque, et dont l'intensité est proportionnelle au changement de portance le long de l'envergure. L'angle d'attaque effectif étant également lié au coefficient de portance du profil d'aile utilisé, il est possible d'en déduire une distribution de portance à partir de ces deux relations sur l'angle d'attaque effectif. La LLT est une méthode efficace qui a cependant quelques limitations : elle fonctionne très mal pour des ailes à faible allongement (grande surface par rapport à l'envergure) et ayant une flèche importante. De plus, étant une méthode 2D, elle prend difficilement en compte un éventuel dièdre de l'aile. Cependant elle est très adaptée pour

2. <http://www.ae.illinois.edu/m-selig/index.html>

- évaluer la traînée visqueuse et le comportement au voisinage des conditions de décrochage (angle d'attaque élevé).
- la méthode VLM (*Vertex Lattice Method*), qui est une alternative linéaire à la LLT, dans le cas notamment où l'aile a un faible allongement, ou un fort dièdre, une large flèche, et même des winglets. La différence provient principalement du fait qu'elle considère la traînée induite comme linéaire et indépendante de la vitesse de l'aile et de l'effet visqueux de l'air. Elle consiste à modéliser la perturbation due à l'aile par une somme de tourbillons distribués le long du plan de l'aile. Leur intensité est calculée de façon à ce qu'ils respectent des conditions aux limites appropriées (conditions de non-pénétration sur la surface des panneaux). Cette méthode doit cependant être utilisée uniquement pour des angles d'attaque assez faibles.
 - la méthode des panneaux 3D, qui permet d'affiner les résultats de la LLT et de la VLM, en prenant notamment en compte l'épaisseur de l'aile, négligée jusqu'ici. Elle permet de donner une idée de la répartition de la pression sur les surfaces supérieure et inférieure de l'aile, ainsi que de fournir une modélisation du fuselage. Son principe est de modéliser la perturbation engendrée par l'aile par une somme de doublets et de sources répartis sur les surfaces supérieure et inférieure de l'aile. Leur intensité doit satisfaire aux conditions aux limites appropriées qui peuvent être du type Dirichlet ou Neumann. Les détails de cette méthode peuvent être trouvés dans [Katz et Plotkin 1991].

Remarque 7.2

Mis à part XFLR5, il existe également d'autres logiciels open-source permettant d'évaluer les coefficients aérodynamiques à partir des méthodes théoriques exposées ci-avant. On peut notamment citer AVL³ ou Tornado⁴ pour Matlab. Ils ont été utilisés en parallèle afin de vérifier la cohérence des résultats obtenus avec XFLR5. Fort heureusement, les trois logiciels ont donné des résultats similaires pour la plupart des coefficients.

7.2.3 Modélisation des élévons

Les élévons sont des surfaces de commande qui combinent les fonctions des ailerons et de la gouverne de profondeur. Ils sont classiquement utilisés sur des profils type aile volante qui ne possèdent pas de stabilisateur à l'arrière de l'avion. Afin de déterminer leur dynamique, les réponses à diverses sollicitations sont obtenues par mesure des tensions en sortie des actionneurs sur le système réel. Tout d'abord, des réponses à des échelons permettent de déterminer les saturations en vitesse de montée et de descente L_{r_u} et L_{r_d} , ainsi que la constante de retard t_d . Puis,

3. <http://web.mit.edu/drela/Public/web/avl/>

4. <http://www.redhammer.se/tornado/>

des sollicitations en rampe donnent la dynamique de l'ensemble qui est modélisée par un système du premier ordre de constante de temps τ . Finalement, les saturations en amplitude supérieure et inférieure L_{m_u} et L_{m_d} sont évaluées en mesurant les braquages maximal et minimal des élévons. Le modèle résultant est comparé à des données expérimentales pour une sollicitation en rampe de L_{m_u} à L_{m_d} sur la figure 7.4. Ceci montre bien la bonne précision du modèle obtenu.

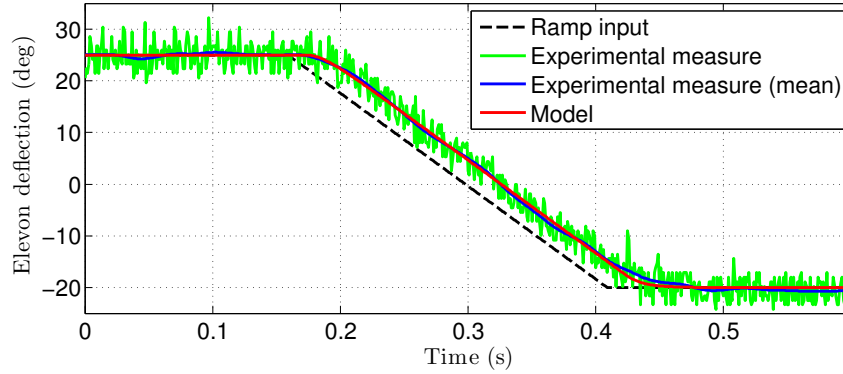


FIGURE 7.4 – Comparaison entre les données expérimentales et le modèle des actionneurs des élévons suite à une sollicitation en rampe.

7.2.4 Modélisation du bloc de propulsion

Modélisation du moteur

La première étape pour modéliser le bloc de propulsion est d'obtenir l'expression de la vitesse de rotation du moteur R_m , exprimée en tours par minute (ou rpm (*revolutions per minute* en anglais)). Cette dernière dépend principalement de la commande en poussée δ_x et de la vitesse air V_a (puisque celle-ci entraîne l'hélice et donc change la vitesse de rotation). δ_x est une valeur sans dimension variant entre 0 (moteur à l'arrêt) et 1 (pleine poussée). Elle possède une dynamique du premier ordre de constante de temps τ_δ , de sorte que la commande filtrée δ_{xf} a pour expression $\delta_{xf} = \frac{1}{1+\tau_\delta s} \delta_x$. Une approximation polynomiale est ensuite utilisée afin de relier R_m à δ_{xf} et V_a :

$$R_m = r_{00} + r_{10}\delta_{xf} + r_{01}V_a + r_{20}\delta_{xf}^2 + r_{11}\delta_{xf}V_a + r_{02}V_a^2 \quad (7.3)$$

Durant des essais en vol, il est possible d'obtenir des valeurs expérimentales de R_m , δ_x et V_a . En utilisant des outils d'approximation polynomiale tels que ceux présents dans la boîte à outils SMAC de Matlab [SMAC 2012-2015] (et détaillés partiellement au chapitre 1 sur l'évaluation d'une LFR à partir de données tabulées), il est possible de déterminer les coefficients r_{ij} de (7.3). Les résultats comparant la vitesse moteur expérimentale avec le modèle sont visualisés sur la figure 7.5. Puisque

l'écart type est de 105 rpm sur 550 secondes de données de vol comparées, on peut affirmer que le modèle est globalement satisfaisant.

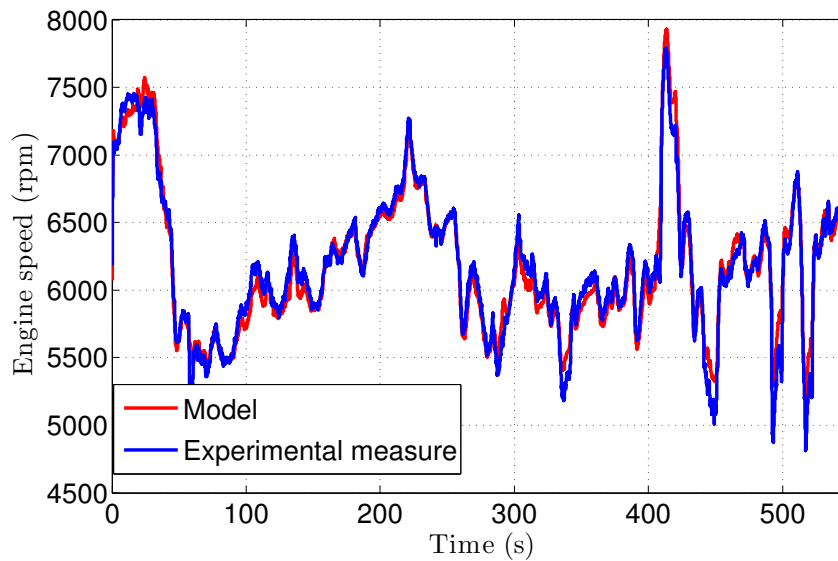


FIGURE 7.5 – Comparaison entre les données expérimentales (bleu) et le modèle (rouge) pour la vitesse moteur R_m .

La figure 7.6 présente finalement la courbe de variation de la vitesse moteur en fonction de V_a et δ_{xf} .

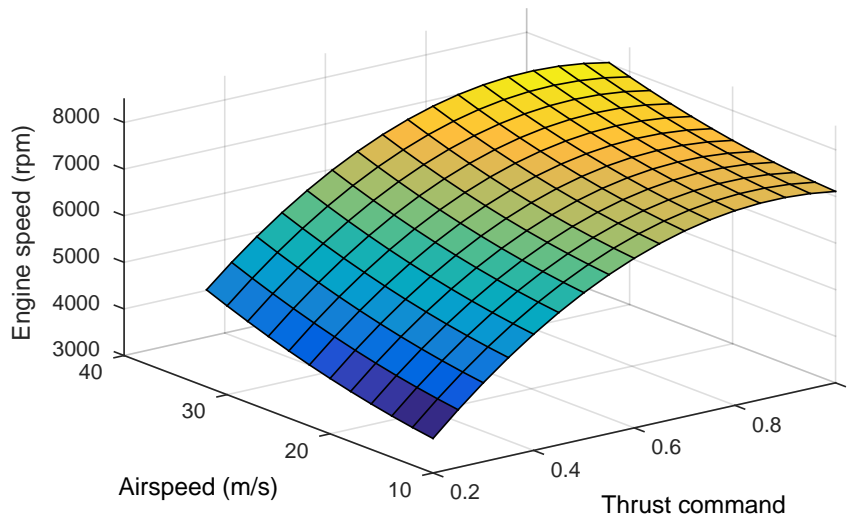


FIGURE 7.6 – Évolution de la vitesse de rotation du moteur (rpm) en fonction de la vitesse air V_a et de la commande de poussée δ_{xf} .

Modélisation de la poussée

Une fois que la vitesse de rotation du moteur est modélisée avec précision, il reste à déterminer un modèle pour la poussée moteur (supposée alignée avec l'axe longitudinal du repère avion, correspondant au terme F_{ex} de (5.20)). La poussée dépend principalement de la vitesse moteur R_m et de la vitesse air V_a , mais contrai-

rement au régime moteur, elle ne peut pas être mesurée en vol. Afin d'obtenir des valeurs numériques de la poussée en fonction de ces deux variables, un logiciel libre tel que QPROP³ peut cependant être utilisé. Ce dernier nécessite de connaître les caractéristiques géométriques de l'hélice (angle et corde des pâles à différents rayons), ses données aérodynamiques (traînée et portance le long du profil peuvent être déterminées en utilisant XFLR5), et les caractéristiques du moteur. Il fournit alors les valeurs de la poussée en fonction de R_m et de V_a sous la forme de données tabulées. Les mêmes outils d'approximation polynomiale qu'avant (bibliothèque APRICOT de SMAC) peuvent être utilisés afin d'en déduire une expression de la forme :

$$F_{ex} = f_{00} + f_{20}R_m^2 + f_{11}R_mV_a + f_{02}V_a^2 \quad (7.4)$$

et d'évaluer les coefficients f_{ij} . La poussée statique peut être mesurée au sol et permet de démontrer que l'approximation précédente avec $V_a = 0$ est satisfaisante, comme le montre la figure 7.7.

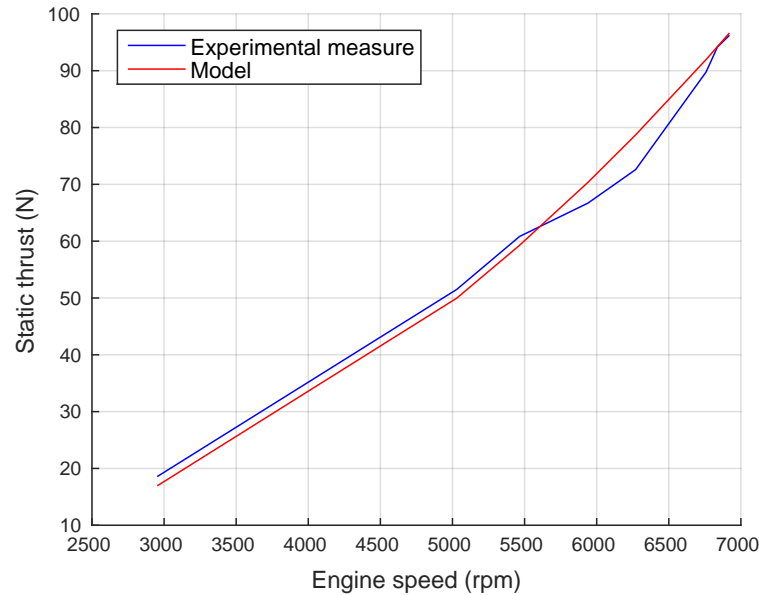


FIGURE 7.7 – Comparaison de la poussée statique ($V_a = 0$) expérimentale (bleu) et modélisée (rouge).

Cependant, il n'y a aucun moyen de valider précisément le modèle lorsque $V_a > 0$ puisqu'aucune donnée expérimentale en vol n'est disponible. Néanmoins, il est possible de vérifier que les trajectoires montantes correspondent bien à une poussée positive. C'est ce que montre la figure 7.8 qui visualise différents couples (R_m, V_a) mesurés en vol. Les points verts et rouges représentent respectivement les phases de montée et de descente de l'avion, et les points encadrés en noir correspondent aux premiers instants des phases montantes. La droite noire continue représente la ligne

3. <http://web.mit.edu/drela/Public/web/qprop/>

de poussée nulle, les points situés en dessous correspondant à une poussée positive, et au-dessus à une poussée négative. On remarque bien que pour la plupart des phases montantes, la poussée est bien positive. De plus, les points correspondants à une phase montante et une poussée négative sont tous des débuts de montée au cours desquels l'avion est entraîné par son élan, ce qui semble tout à fait cohérent. Cette validation, certes approximative et très empirique, permet cependant d'avoir relativement confiance dans le modèle obtenu.

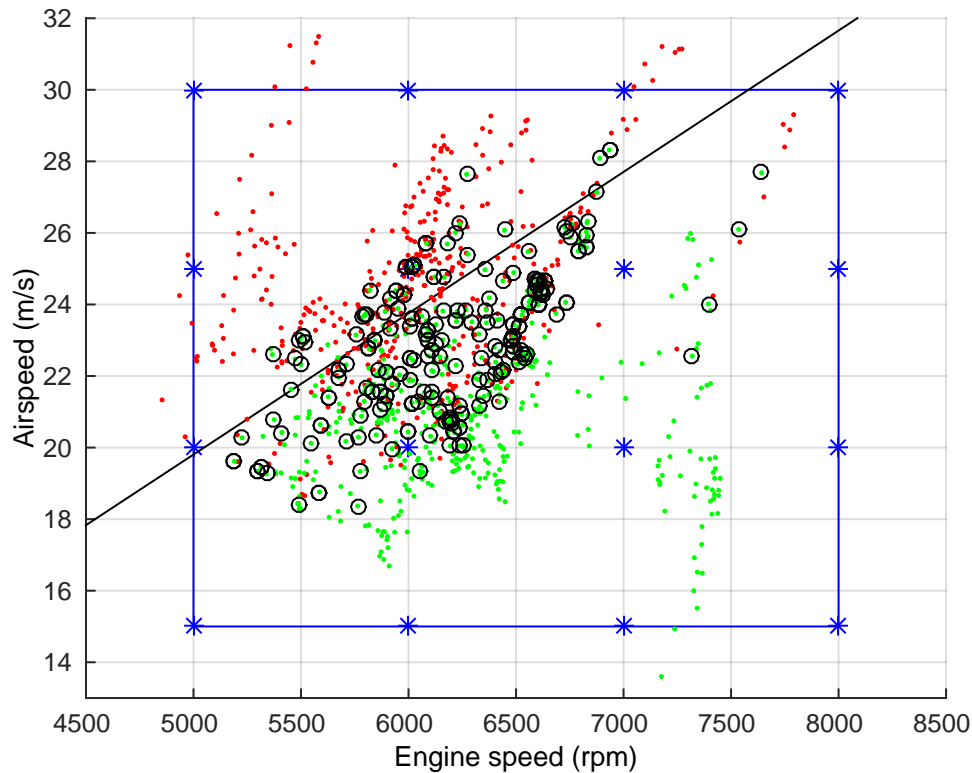


FIGURE 7.8 – Données expérimentales permettant de valider partiellement le modèle de poussée en vol. Les points rouges correspondent à des phases de vol descendantes, les points verts à des phases ascendantes, dont les premiers instants sont mis en évidence par des cercles noirs, et le trait noir continu à la zone de poussée nulle.

Enfin la figure 7.9 montre bien que la poussée F_{ex} telle que modélisée par (7.4) est une fonction croissante de la vitesse de rotation R_m et décroissante de la vitesse air V_a , ce qui est cohérent.

7.2.5 Simulations en boucle ouverte et comparaison avec des données de vol

Afin de valider le modèle complet, dont toutes les valeurs numériques ont été facilement implantées au sein du modèle Simulink générique d'avion proposé par SMAC [SMAC 2012-2015] (et présenté dans la conclusion du chapitre 5), des simulations en boucle ouverte ont été réalisées. Les consignes δ_x , δ_a et δ_e envoyées par le pilote lors d'essais en vol sont injectées dans le modèle et les sorties sont comparées avec les données mesurées en vol. On se concentre ici sur une phase assez longue

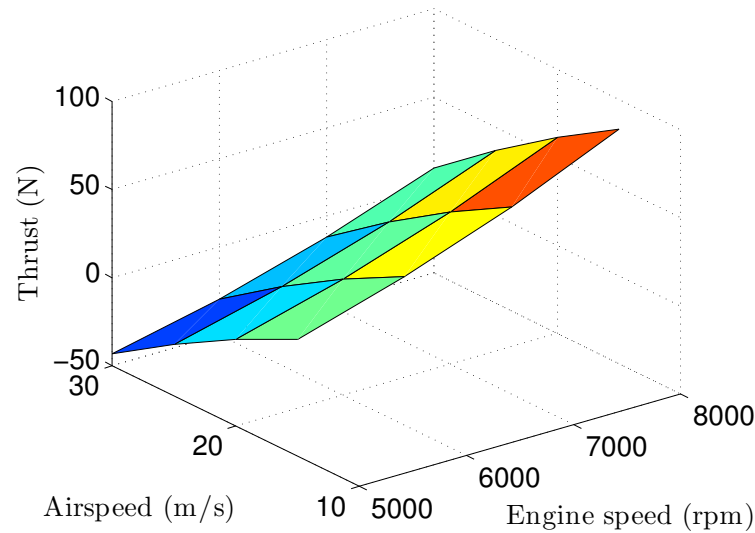


FIGURE 7.9 – Évolution de la poussée moteur en fonction de la vitesse de rotation et de la vitesse air.

de 30 secondes, et sur deux phases courtes où des échelons sur les entrées ont été réalisés (en longitudinal et en latéral) pour comparer à la fois le comportement dynamique long terme et court terme. Chaque ensemble de courbes est regroupé sur deux figures : la première présente les variables longitudinales (angle de tangage θ , vitesse air V_a , vitesse de tangage q et déflexion des gouvernes de profondeur δ_e), et la deuxième les variables latérales (angle de roulis ϕ , vitesse de roulis p , déflexion des ailerons δ_a) avec également la vitesse de rotation du moteur R_m . Les données expérimentales sont en bleu, et les simulées en vert.

Remarque 7.3

Quelques remarques sont à introduire avant de présenter les comparaisons :

- Pour une configuration donnée, l'avion modélisé sur Simulink est d'abord trimé afin de trouver les valeurs de δ_e , δ_x , et δ_a permettant un maintien au point de vol considéré. La routine de trim est basée sur un processus dichotomique afin de déterminer toutes les grandeurs à l'équilibre. Il faut noter que ces valeurs d'équilibre ne sont pas forcément identiques à celles de l'avion réel en vol à un instant donné. Ce dernier est en effet sujet aux perturbations (non modélisées), et possède également une géométrie qui peut être légèrement différente – l'avion modélisé étant notamment idéal et sans défaut. Ainsi, seules les variations des consignes pilotes sont envoyées au modèle de simulation, et non les valeurs brutes. Les valeurs de δ_e , δ_x , et δ_a ne sont donc pas identiques en vol et en simulation, mais leurs variations le sont, permettant de comparer effectivement la dynamique.
- Il existe un écart non expliqué sur l'angle de roulis ϕ , ce qui semble surprenant étant donné que la vitesse de roulis p est cohérente entre les simulations et les données de vol. Aucune explication ne peut être donnée au moment de l'écriture de ce mémoire quant à la validité des mesures expérimentales d'angle de roulis,

- Enfin il faut souligner que toutes les comparaisons ont été réalisées en boucle ouverte, et que le modèle ne prend pas en compte les perturbations atmosphériques (qui ne sont pas mesurables), ce qui peut amener des écarts plus ou moins importants entre les courbes. Des essais en boucle fermée permettront à priori de réduire les effets de ces perturbations et de mieux valider le modèle.

Simulations à moyen terme : 30 secondes de vol

Des comparaisons obtenues sur 30 secondes de vol sont présentées sur les figures 7.10 et 7.11. La dynamique et les valeurs des différents paramètres restent cohérentes, même s'il est difficile de conclure sur un temps aussi élevé. En effet, les perturbations atmosphériques comme le vent peuvent varier de façon considérable sur un tel horizon de temps et affectent nécessairement le comportement de l'avion réel en vol. Elles ne peuvent cependant pas être mesurées et ne sont pas prises en compte lors de la simulation du modèle. Ce dernier semble toutefois satisfaisant dans l'ensemble.

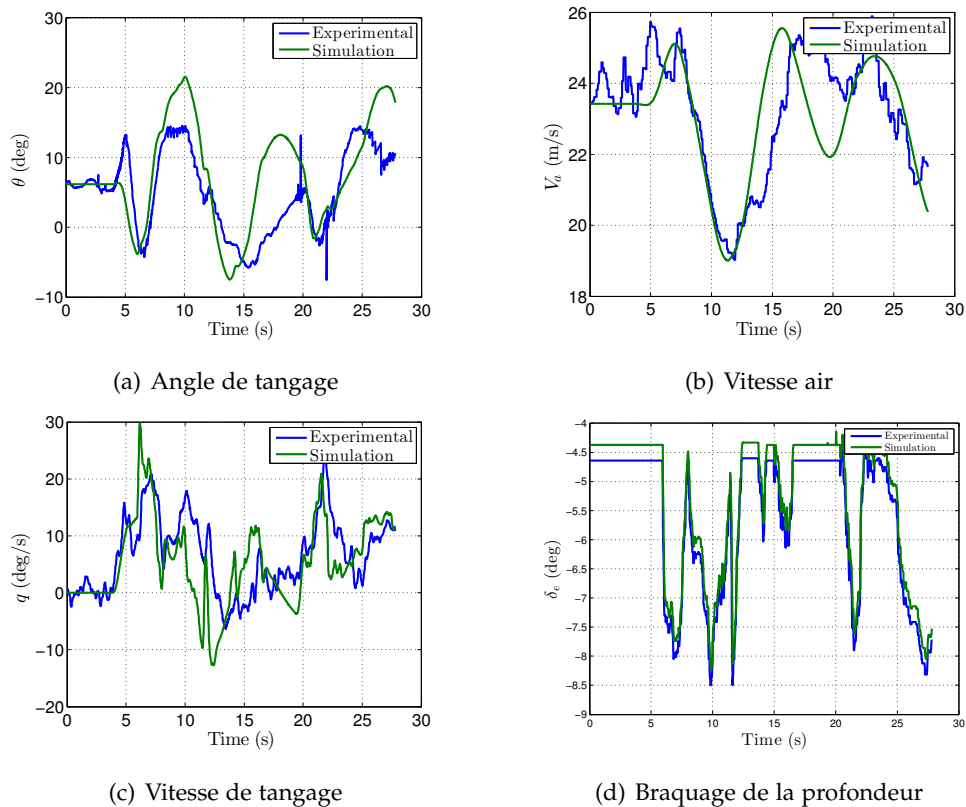


FIGURE 7.10 – Comparaison sur 30 secondes de données simulées et de vol : modèle longitudinal.

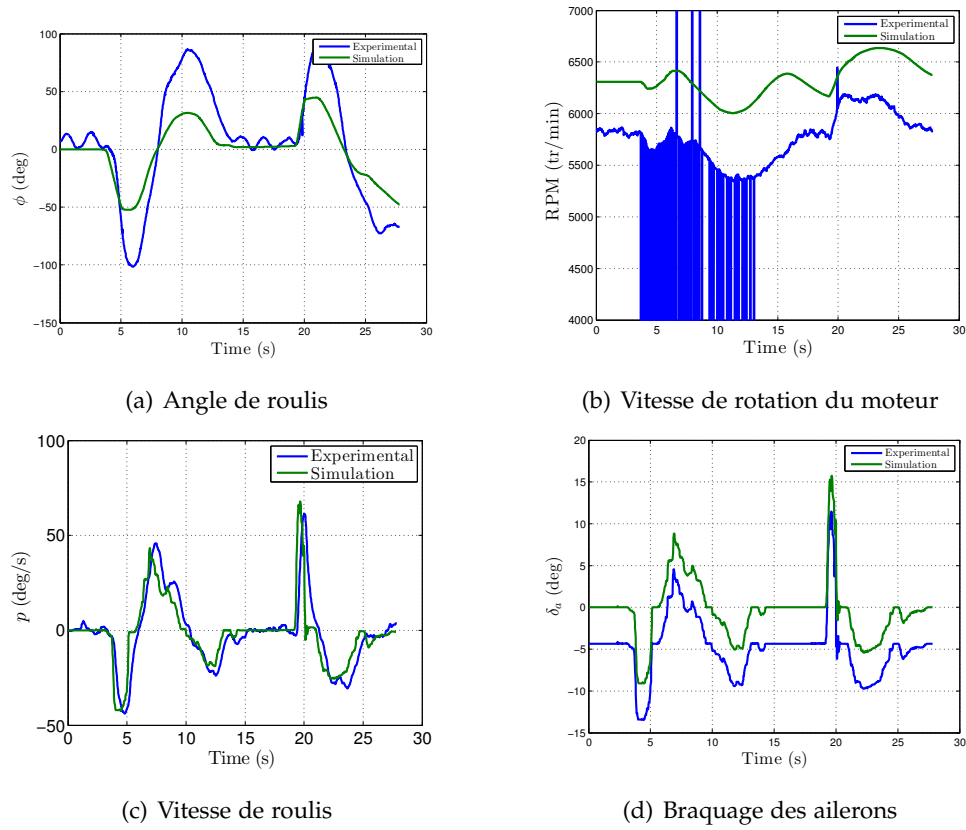


FIGURE 7.11 – Comparaison sur 30 secondes de données simulées et de vol : modèle latéral.

Simulations à court terme : échelon longitudinal

Un échelon sur δ_e est réalisé (voir figure 7.12(d)) et les comportements associés sont représentés sur les figures 7.12 et 7.13. La dynamique court terme est bien respectée, particulièrement sur la vitesse air. Les variables latérales expérimentales ne sont pas du tout constantes alors qu'elles ne sont pas sollicitées par le pilote. Ceci montre bien qu'il devait exister des perturbations atmosphériques (vent, turbulences) qui peuvent rendre difficile la comparaison des données et l'évaluation de la précision du modèle.

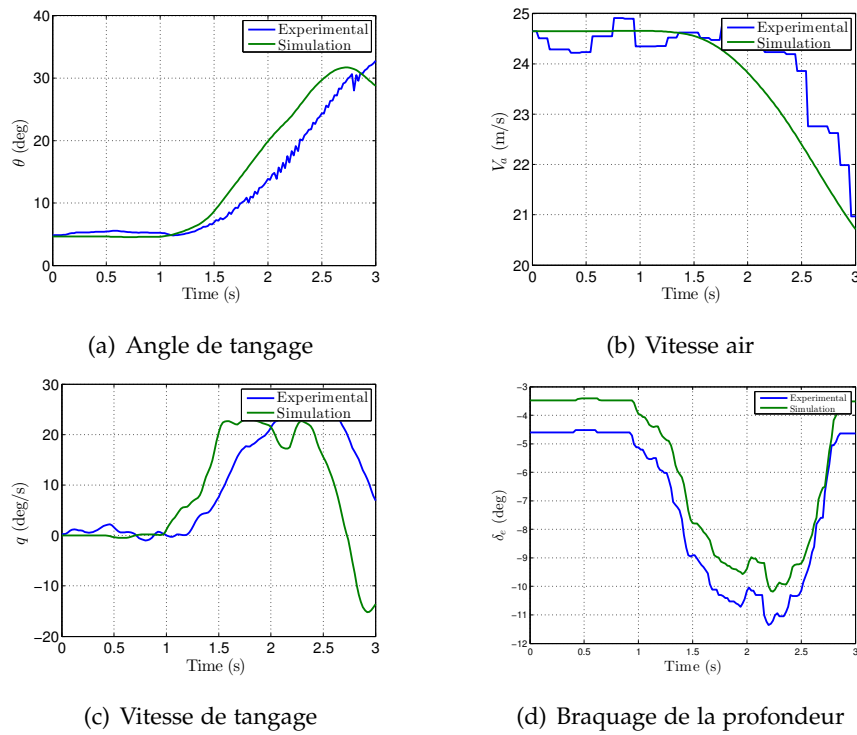


FIGURE 7.12 – Comparaison de données simulées et de vol lors d'un braquage de la profondeur : modèle longitudinal.

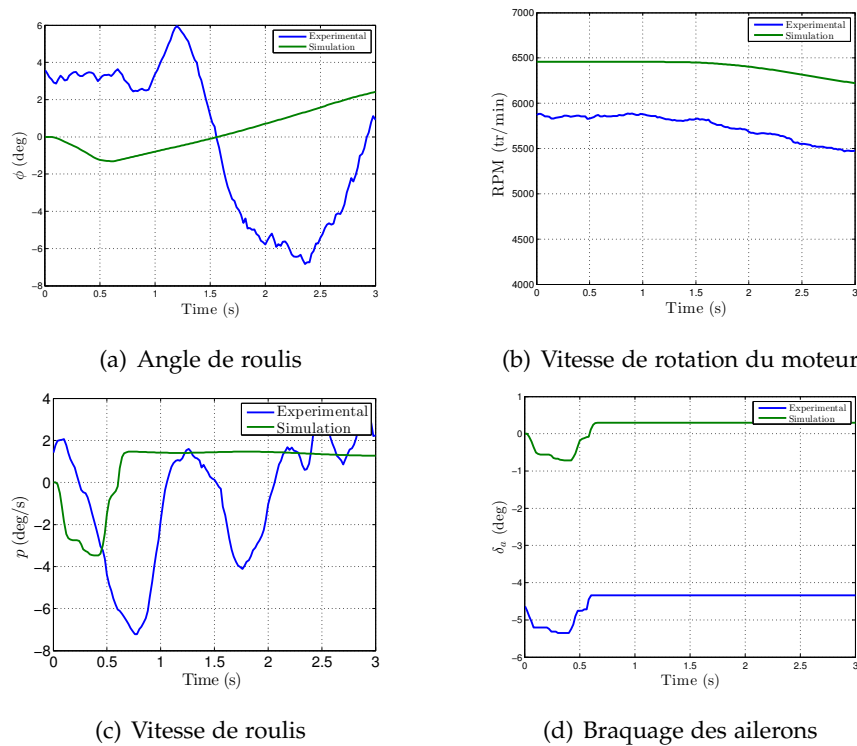


FIGURE 7.13 – Comparaison de données simulées et de vol lors d'un braquage de la profondeur : modèle latéral.

Simulations à court terme : échelon latéral

Un échelon sur δ_a est réalisé et les comportements associés sont représentés sur les figures 7.14 et 7.15. Comme précisé à la remarque 7.3, un écart inexplicable est observé sur l'angle de roulis (figure 7.15(a)), qui présente des valeurs expérimentales manifestement excessives (car supérieures à 90deg en valeur absolue). Ceci est d'autant plus surprenant que les valeurs de la vitesse de tangage p sont quant à elles cohérentes (figure 7.15(c)).

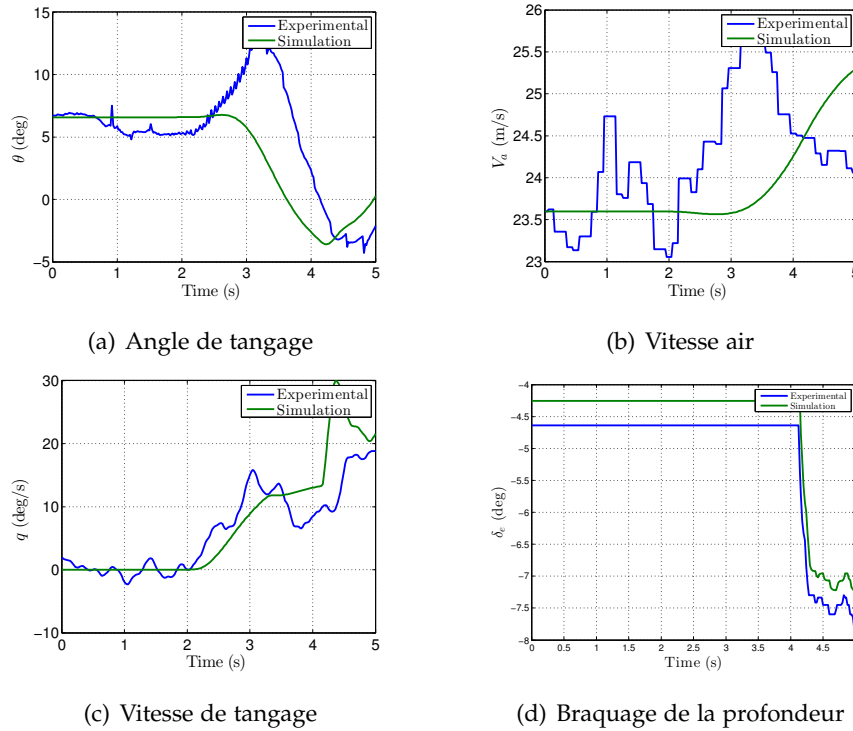


FIGURE 7.14 – Comparaison de données simulées et de vol lors d'un braquage des ailerons : modèle longitudinal.

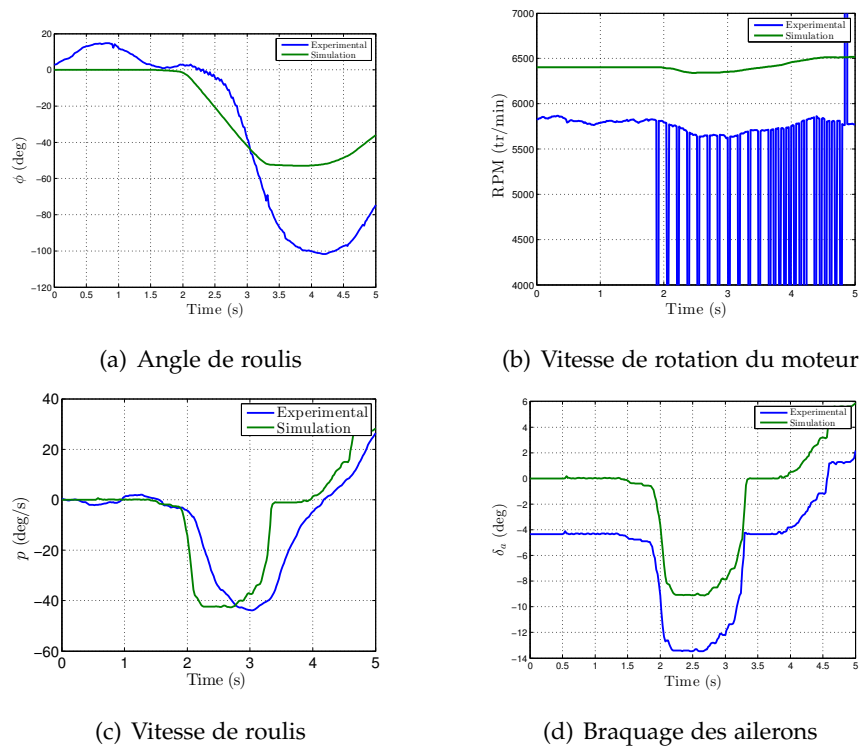


FIGURE 7.15 – Comparaison de données simulées et de vol lors d'un braquage des ailerons : modèle latéral.

7.3 SYNTHÈSE DE LOIS DE PILOTAGE AUTOMATIQUE

Cette section propose de synthétiser une loi de pilotage longitudinal par deux méthodes :

- la première, classique, correspond à un placement de pôles et un placement de structure propre, et sert de base pour la suite,
- la deuxième est celle proposée au chapitre 4 de ce mémoire, afin de montrer qu'elle est facilement applicable au pilotage d'un drone à voilure fixe et particulièrement adaptée aux contraintes de découplage, au rejet de perturbations, et à la prise en compte d'incertitudes.

L'objectif est de piloter la vitesse air V_a et l'angle de vol γ pour différentes configurations de vol et en présence d'incertitudes de modélisation. Le modèle longitudinal considéré, issu des équations de la mécanique du vol rappelées au chapitre 5, est celui décrit par le système d'équations (6.1).

7.3.1 Synthèse modale

La synthèse modale est une technique linéaire, il faut donc dans un premier temps obtenir un modèle linéarisé de l'avion autour d'une condition d'équilibre. La configuration nominale est choisie pour des valeurs centrales des paramètres

variants (masse, position du centre de gravité et vitesse, sans aucune incertitude paramétrique). La méthode de placement de pôles permet ensuite d'obtenir un correcteur longitudinal noté K_{mod} , qui commande la poussée δ_x et le braquage de la profondeur δ_e , et possède 6 entrées telles que :

$$\begin{bmatrix} \delta_x \\ \delta_e \end{bmatrix} = K_{mod} \begin{bmatrix} \int (V_{ac} - V_a) & \int (\gamma_c - \gamma) & V_a & \gamma & q & \theta \end{bmatrix}^T \quad (7.5)$$

Les intégrateurs permettent d'assurer la précision en régime permanent, tandis que V_{ac} et γ_c désignent les consignes en vitesse et en pente. La structure de la loi est très proche de celle du correcteur de la boucle externe de la méthode de compensation non-linéaire robuste du chapitre 4, sans les entrées correspondant aux perturbations non-linéaires mesurées. Elle permet donc d'avoir une idée de la réponse fréquentielle du système bouclé afin de dimensionner les pondérations pour la synthèse \mathcal{H}_∞ réalisé dans la section 7.3.2.

Cette loi de commande modale offre 12 degrés de liberté, 6 permettant de placer les pôles en boucle fermée, et les 6 autres permettant d'assurer le découplage. Le tableau 7.1 présente les valeurs des pôles en boucle ouverte et en boucle fermée. Le but premier est d'accélérer et d'amortir le mode phugoïde de l'avion, qui autrement entraîne des oscillations lentes et mal amorties. L'oscillation d'incidence ne doit pas être trop ralentie. Il faut noter que les dynamiques des élévons et du moteur ne peuvent pas être imposées, mais la valeur des pôles placés ne les dégrade pas de façon critique. Les simulations temporelles seront présentées à la sous-section 7.3.3.

Mode	Boucle ouverte	Boucle fermée
Phugoïde	$-0.016 \pm 0.744i$	$-0.700 \pm 0.420i$ (placés)
Oscillation d'incidence	$-5.070 \pm 8.280i$	$-3.500 \pm 2.625i$ (placés)
Intégrateurs	0 ; 0	$-4.000 \pm 3.000i$ (placés)
Moteurs	-2.500	-1.014
Élevons	-76.923	-72.182

TABLEAU 7.1 – Modes en boucle ouverte et en boucle fermée avec le correcteur modal.

7.3.2 Technique de compensation non-linéaire robuste

La technique de compensation non-linéaire robuste, développée de façon théorique au chapitre 4, est ici appliquée au modèle longitudinal du drone avion. La méthode étant extrêmement générique pour le modèle d'avion développé, la seule étape qui va nécessiter un peu d'adaptation par rapport au modèle d'avion civil du chapitre 6 est la phase de dimensionnement des pondérations fréquentielles pour

la synthèse \mathcal{H}_∞ (équations (6.9)-(6.11)). Le lecteur est invité à parcourir la section 6.1 page 131 du chapitre précédent pour plus d'informations.

La boucle interne de linéarisation est en effet inchangée, et concernant la boucle externe, les pondérations suivantes, du même format que pour l'avion civil, sont choisies pour la synthèse \mathcal{H}_∞ (voir le schéma de synthèse de la figure 4.1 page 99) :

$$W_p(s) = \begin{pmatrix} \frac{s/2+0.5}{s+5 \times 10^{-3}} & 0 \\ 0 & \frac{s/2+2.35}{s+2.35 \times 10^{-4}} \end{pmatrix} \quad (7.6)$$

$$W_u(s) = \begin{pmatrix} \frac{s+0.01}{0.001s+40} & 0 \\ 0 & \frac{s+0.04}{0.001s+75} \end{pmatrix} \quad (7.7)$$

Il est important de noter que le correcteur modal a été utilisé afin de déterminer rapidement les gabarits fréquentiels souhaités en boucle fermée. Quant au filtre $F(s)$, il est choisi comme suit :

$$F(s) = \begin{pmatrix} \frac{1}{0.7s+1} & 0 & 0 & 0 \\ 0 & \frac{1}{0.7s+1} & 0 & 0 \\ 0 & 0 & \frac{1}{0.7s+1} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (7.8)$$

Le modèle de référence $R(s)$ (voir équation (6.7) page 133) est également adapté compte tenu des capacités dynamiques de l'avion évaluées grâce aux essais en vol. Il est désormais donné par :

$$R(s) = \begin{bmatrix} \frac{1}{1.5s+1} & 0 \\ 0 & \frac{2.4^2}{(0.1s+1)(s^2+2*0.75*2.4s+2.4^2)} \end{bmatrix} \quad (7.9)$$

et on a bien une dynamique du premier ordre en vitesse et du troisième ordre en angle de vol.

Remarque 7.4

Il est intéressant de remarquer à ce stade que seuls les filtres et les modèles de référence ont dû être adaptés, et ce simplement en se fiant à la dynamique du drone. Ceci montre bien le caractère très générique de la méthode de synthèse lorsqu'il s'agit de dimensionner des lois pour un avion rigide.

L'algorithme de synthèse \mathcal{H}_∞ structurée, d'ordre fixé à 3, multi-modèles avec extraction automatique de pire-cas par μ -analyse est exécuté, ce qui permet d'obtenir en quelques itérations une boucle externe robuste qui stabilise l'ensemble des

configurations admissibles en présence d'incertitudes sur les coefficients aérodynamiques. On notera $K(s)$ ce correcteur par la suite.

7.3.3 Simulations temporelles

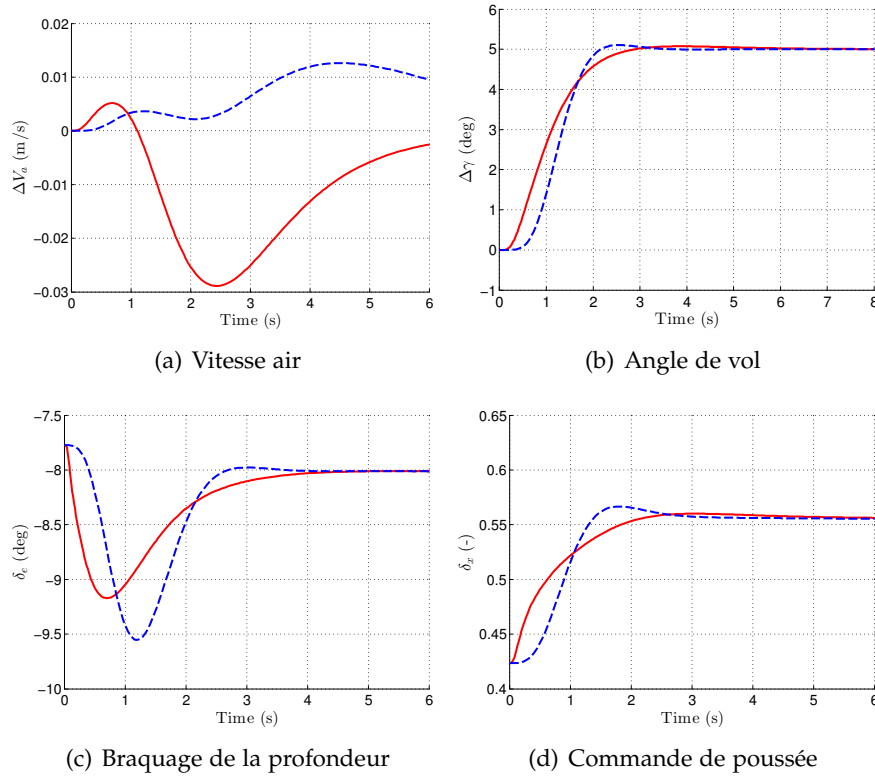


FIGURE 7.16 – Simulations sur le modèle non-linéaire complet pour une consigne échelon de 5 degrés sur γ , avec le correcteur modal (ligne en tirets bleus) et la loi non-linéaire robuste multi-modèles (trait plein rouge), pour la configuration nominale **sans incertitudes**.

Des simulations temporelles sur le modèle non-linéaire complet du drone sont exécutées pour une consigne échelon de 5 degrés sur l'angle de vol, à la fois pour le système bouclé avec le correcteur nominal K_{mod} et pour celui avec la loi non-linéaire robuste (boucle interne linéarisante et boucle externe robuste avec le correcteur $K(s)$).

Deux figures sont proposées. La figure 7.16 présente les réponses pour la configuration nominale (celle pour laquelle le correcteur modal a été dimensionné), sans aucune incertitude de modélisation. Les performances entre les deux lois de pilotage sont très similaires dans ce cas-ci. Il est pourtant bien évident qu'en présence d'incertitudes de modélisation, ou pour d'autres configurations paramétriques, la loi modale doit voir ses performances dégradées puisque rien n'a été fait pour les prendre en compte.

La figure 7.17 présente la configuration "quasi" pire-cas, évaluée par μ -analyse (la pire-cas déstabilise complètement la loi modale, ce qui ne permet plus d'observer

son comportement sur les courbes). Ceci montre clairement l'intérêt de la technique de synthèse développée dans ce mémoire qui permet de garder un niveau de performances satisfaisant quel que soit la configuration paramétrique du drone.

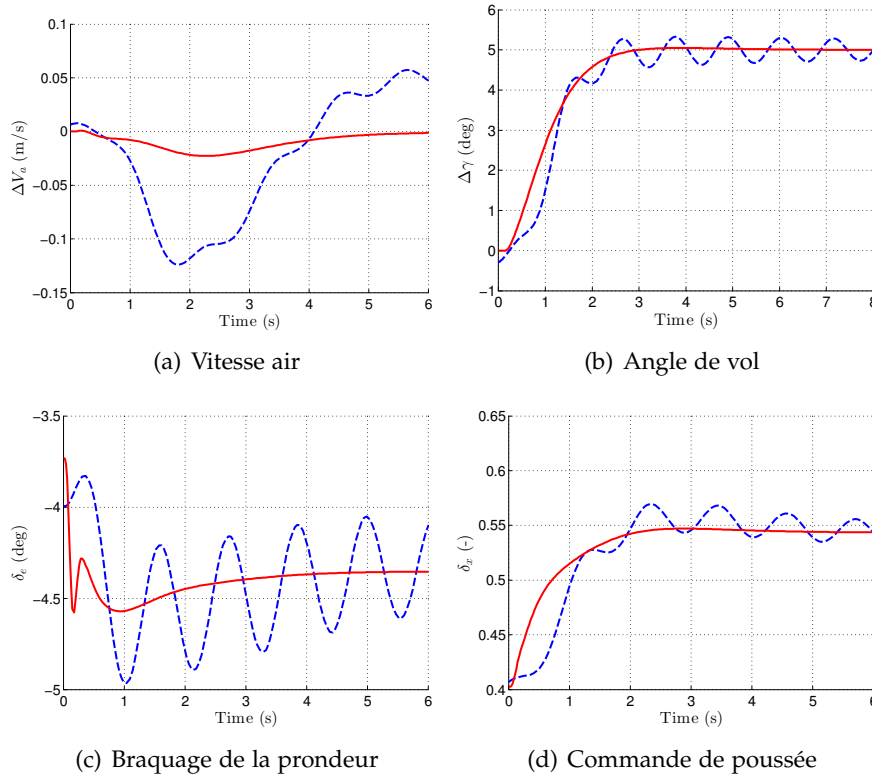


FIGURE 7.17 – Simulations sur le modèle non-linéaire complet pour consigne échelon de 5 degrés sur γ , avec le correcteur modal (ligne en tirets bleus) et la loi non-linéaire robuste multi-modèles (trait plein rouge), pour la configuration pire-cas **avec incertitudes**.

7.4 DE L'IMPLANTATION RÉELLE DES LOIS

Cette dernière section vise à donner quelques informations sur la possibilité d'implanter sur l'avion réel la loi de pilotage non-linéaire robuste (la loi modale est bien évidemment facilement implantable, puisqu'elle ne contient que des gains statiques). Le micro-processeur utilisé sur l'Avion Jaune est un ARM[®] Cortex[®]-M4⁴ composé de deux cœurs de fréquence 80Mhz. Les opérations mathématiques de la boucle interne de la loi peuvent être réalisées sans ajout de librairie supplémentaire, puisqu'il dispose d'une unité de calcul de flottants intégrée. Il y a une seule matrice à inverser (la matrice Λ de l'équation (6.6) page 132) qui est de taille 2×2 sans aucune singularité, ce qui ne pose aucun soucis particulier (il existe une expression analytique de l'inverse bien connue...). L'ordre du correcteur de la boucle externe est limité à 3 ce qui, encore une fois, est facilement traité par

4. <http://www.arm.com/products/processors/cortexm/cortex-m4-processor.php>

le processeur. Ceci permet de bien montrer l'intérêt pratique d'utiliser la version structurée de la synthèse \mathcal{H}_∞ , puisqu'autrement, l'ordre aurait été bien plus élevé ce qui aurait rendu difficile l'implantation.

Concernant le déroulement des tâches de calculs, le processeur opère avec une période de 20ms, et la place disponible pour la partie commande dans ce laps de temps est actuellement de 10ms, lorsqu'on enlève les autres tâches telles que l'acquisition, le filtrage, ou la fusion de données. Étant donné la complexité numérique assez faible de la loi comme vu précédemment, ceci est largement suffisant. Enfin, il convient également de préciser que la loi est robuste aux retards jusqu'à $50\text{ms} > 20\text{ms}$, suite à des simulations qui ont été réalisées pour différentes valeurs de discrétisation. Il est donc parfaitement possible d'implanter la loi de commande non-linéaire robuste afin de réaliser des essais en vol en boucle fermée.

Remarque 7.5

A la date d'écriture de ce mémoire, il n'a pas encore été possible d'implanter la loi longitudinale. Il manque sur l'avion des capteurs d'angle d'incidence et de vitesse air vraie, ce qui pose problème étant donné que la loi a besoin de mesures précises de ces deux données. Récemment, une solution a été trouvée dans le commerce dont le principe réside en un tube de Pitot avec des ailettes montées autour d'un axe pivotant et reliées à un capteur d'angle analogique. Le tube de Pitot est alors parfaitement dans le flux d'air, et l'angle d'incidence est déduit du capteur d'angle. Après installation de ces capteurs, des essais en vol pourront être effectués, et il sera dans un futur proche possible d'implanter la loi de pilotage pour des essais.

CONCLUSION DU CHAPITRE

Ce chapitre a proposé une méthodologie complète pour modéliser un drone à voilure fixe, en se basant sur les équations générales de la mécanique du vol qui ont été exposées au chapitre 5. La méthodologie est suffisamment générique pour être facilement adaptée à tout type de drone avion. A titre d'exemple, il a été proposé d'appliquer la méthode à l'Avion Jaune disponible au sein de l'ONERA, un avion type aile volante de 4 mètres d'envergure. Le modèle a été comparé à des données d'essais en vol et semble perfectible mais largement satisfaisant à l'heure actuelle dans une perspective de synthèse de lois de pilotage. Il serait judicieux de le peaufiner par des essais en boucle fermée afin de réduire l'effet des perturbations et également par des méthodes d'identification, d'autant plus que beaucoup de données sont mesurables grâce à l'acquisition récente de nouveaux capteurs.

Il sera également intéressant par la suite de tester sur l'avion réel la méthode de

synthèse de loi non-linéaire robuste (chapitre 4). Elle a été facilement adaptée par rapport au modèle d'avion civil (chapitre 6)) puisqu'assez générique. Les capacités du micro-processeur embarqué permettent de plus son implantation. Ce chapitre s'est consacré au développement des lois du modèle longitudinal, ce dernier ayant été montré plus fiable que son analogue latéral, mais un correcteur latéral a également été développé de façon similaire. Il est d'ordre 2 et permet de commander l'angle de roulis ϕ grâce aux braquages des ailerons δ_a . L'angle de dérapage β n'est pour le moment pas commandable puisqu'aucune dérive n'est pour le moment installée sur le drone.

Pour finir, l'auteur souhaite tout particulièrement adresser ses remerciements à Paul Chavent pour l'aide fournie sur la modélisation de l'avion à l'aide de XFLR5, Pierre Escalas pour les essais en vol, Alexandre Amiez pour les informations concernant l'implantation, et Mathieu Rognant pour la modélisation sous CATIA.

CONCLUSION GÉNÉRALE

Cette thèse est consacrée au développement de méthodes de synthèse et de validation de lois de commande non-linéaires pour le pilotage robuste d'un avion sur l'ensemble de son domaine de vol en présence d'incertitudes paramétriques. Elle apporte des contributions tant au niveau des méthodes de synthèse qu'au niveau de l'analyse de robustesse des systèmes LTI.

RÉSUMÉ DES CONTRIBUTIONS

Sur le plan théorique : analyse de robustesse

Tout d'abord, la première partie s'est intéressée aux notions bien connues d'analyse de robustesse des systèmes incertains LTI. Après quelques rappels sur la modélisation LFR et la μ -analyse, le chapitre 1 a présenté sous forme d'état de l'art les principales méthodes de calcul des bornes de la valeur singulière structurée μ , dont le calcul exact est en effet un problème NP-difficile. Que ce soit pour les bornes inférieure ou supérieure, de nombreuses techniques ont émergé durant les trois dernières décennies. Elles peuvent être de complexité polynomiale ou exponentielle, fortement ou peu conservatives, applicables sur un maillage fréquentiel ou sur une plage de fréquences. Pour des applications réalistes, il existe des techniques de calcul de borne inférieure de μ très efficaces. En ce qui concerne la borne supérieure cependant, trois principaux problèmes ont été identifiés :

- les techniques à temps polynomial, notamment l'approche classique par matrices (D,G), sont parfois trop conservatives,
- le temps de calcul pour certaines méthodes moins conservatives est prohibitif,
- il n'est pas toujours possible de se passer d'un maillage fréquentiel pour calculer une borne supérieure de μ .

Le chapitre 2 a apporté des contributions sur ces trois points. Les μ -sensibilités ont été introduites pour évaluer les incertitudes à considérer en priorité afin de réduire le nombre de contraintes LMIs lorsqu'il s'agit d'utiliser des méthodes peu conservatives mais numériquement complexes, ou bien d'améliorer la technique classique de branch-&-bound en subdivisant le domaine d'incertitudes de manière pertinente, réduisant fortement le nombre d'itérations. De plus, pour l'approche

par multiplieurs, une technique de type Hamiltonien a été développée afin de se passer de maillage fréquentiel et permettre ainsi une validation de la borne sur des intervalles de fréquence. L'efficacité de chacune des techniques a été prouvée sur un ensemble très complet de benchmarks réalistes. Même dans les cas difficiles, des marges de robustesse précises ont été obtenues avec une bonne maîtrise des temps de calcul.

Sur le plan théorique : synthèse de lois

La deuxième partie de ce mémoire est consacrée à la synthèse. L'objectif a été de développer une loi de commande permettant de faire face à un comportement dynamique non-linéaire d'un véhicule évoluant au sein d'un large domaine de fonctionnement, dont le modèle comprend des incertitudes. Une technique intéressante, notamment pour des applications aéronautiques, est l'inversion dynamique. Elle a été présentée au chapitre 3, avec également des rappels sur la synthèse \mathcal{H}_∞ qui permet d'apporter une garantie de robustesse de la boucle externe. Le problème persistant est cependant le réel manque de robustesse de l'ensemble, d'une part parce que le modèle linéarisé par la boucle interne est sous forme canonique avec une série d'intégrateurs (forme de Brunovsky), faisant perdre ainsi la connaissance sur la dynamique du système initial. D'autre part, les incertitudes de modélisation ne sont pas prises en compte explicitement lors de la synthèse.

Le chapitre 4 propose de contrer les défauts principaux du schéma d'inversion dynamique robuste classique cités précédemment, et constitue donc la deuxième contribution théorique et méthodologique de cette thèse. Basée sur les techniques d'inversion dynamique, une nouvelle approche est proposée. La structure de la loi développée est classiquement constituée de deux boucles imbriquées :

- La boucle interne permet de linéariser partiellement le modèle non-linéaire, aboutissant à un nouveau système constitué non plus d'une série d'intégrateurs, mais plutôt du Jacobien du système non-linéaire de départ. L'avantage principal est de pouvoir garder le plus d'informations possibles sur la dynamique du système initial. La boucle interne permet donc de gérer les principales non-linéarités en les compensant, tout en adaptant la loi au point de fonctionnement considéré.
- La boucle externe est synthétisée sur le modèle partiellement linéarisé obtenu, et utilise la synthèse \mathcal{H}_∞ structurée. L'originalité de l'approche est double. D'une part, la boucle externe est construite de façon à rejeter les non-linéarités restantes (non compensées par la boucle interne), ces dernières étant estimées et directement accessibles en entrée du correcteur. Ceci permet d'étendre le domaine admissible et donc de traiter le large domaine opérationnel du sys-

tème. D'autre part, l'autre originalité concerne l'utilisation des outils d'analyse de robustesse – d'où le lien avec la première partie de ce mémoire – pour évaluer des pire-cas et les prendre en compte lors du calcul du correcteur grâce aux possibilités multi-modèles de la synthèse \mathcal{H}_∞ structurée. Un algorithme itératif de synthèse / validation est donc proposé, permettant d'obtenir un correcteur robuste stabilisant toutes les configurations du domaine admissible. Grâce aux contributions de la première partie permettant une évaluation précise de la marge de stabilité dans le cas LTI, il n'y a pas de risques de sous-estimer la robustesse du correcteur⁵, ce qui aurait entraîné des itérations de synthèse supplémentaires et un conservatisme accru de celui-ci.

Sur le plan applicatif

Ce travail de thèse est principalement axé sur les applications aéronautiques. Celles-ci ont été exposées à la troisième partie du manuscrit, et les méthodes développées de synthèse de lois de pilotage se sont avérées particulièrement bien adaptées. Le comportement assez fortement non-linéaire d'un avion, sur un domaine de fonctionnement relativement large, permet en effet aux techniques basées sur l'inversion dynamique robuste de s'imposer comme des alternatives intéressantes aux méthodes de séquençement de gain.

Après quelques rappels sur les équations générales de la mécanique du vol au chapitre 5, les chapitres 6 et 7 détaillent une application complète de la méthode de synthèse respectivement sur un modèle d'avion civil puis sur un modèle de drone à voilure fixe. En présence d'incertitudes et de paramètres variants, la méthode se révèle très efficace pour le pilotage robuste et performant de ces engins.

Plus spécifiquement, le chapitre 6 s'est concentré sur le développement de lois de guidage pour l'atterrissage automatique complet d'un avion de transport de passagers par suivi d'ILS, permettant de plus l'arrondi et le décrochage en cas de vent latéral. Des simulations de Monte Carlo ont permis de constater que l'ensemble guidage+commande satisfait à des contraintes réalistes.

Le chapitre 7 s'est quant à lui concentré sur la présentation d'une méthode générique pour la modélisation d'un drone à voilure fixe, qui a permis d'obtenir un modèle réaliste validé par des essais en vol. L'intérêt d'une telle application réside dans la possibilité d'implanter facilement la loi de pilotage obtenue sur le système réel en vue d'essais en vol en boucle fermée.

5. D'autant plus que les incertitudes rencontrées en pratique, en particulier dans notre contexte d'inversion dynamique, sont souvent LTV.

CRITIQUES ET PERSPECTIVES

Analyse de robustesse des systèmes LTI

Concernant l'analyse de robustesse, ces travaux de thèse se sont principalement attardés sur la réduction du conservatisme des bornes de μ . Bien que pertinentes et efficaces en pratique, il apparaît clairement que les contributions utilisant les μ -sensibilités restent plutôt heuristiques, et demeurent fortement dépendantes à l'heure actuelle des capacités de calcul et des outils utilisés notamment pour la résolution de LMIs (type de solveur). La μ -analyse est étudiée depuis de nombreuses années, et il devient de plus en plus difficile de développer de nouvelles méthodes à la fois peu conservatives et utilisables sur des applications réalistes. Évoquer comme perspectives de trouver de nouvelles pistes et méthodes encore plus efficaces semble ainsi très ambitieux. Il est enfin clair que la classe de systèmes considérée ici peut sembler assez limitée, puisque bornée uniquement aux dynamiques négligées et incertitudes paramétriques invariantes dans le temps. Mais elle n'en reste pas moins d'actualité pour de nombreuses applications industrielles, évitant en partie les coûteuses simulations de Monte Carlo.

Technique de compensation non-linéaire robuste

La méthode de synthèse développée dans ces travaux de thèse présente des résultats intéressants tant du point de vue de sa facilité de mise en œuvre et de son aspect générique, que des niveaux de performance et de robustesse qu'elle offre. Elle s'avère bien adaptée au pilotage des engins aéronautiques, mais elle pourrait très bien s'appliquer à d'autres classes de systèmes. La principale critique et limitation de l'approche concerne la partie analyse du schéma itératif. Le modèle d'analyse de robustesse utilisé est en effet fortement conservatif, puisque ne considérant pas la structure des non-linéarités, prises comme des perturbations extérieures. Celles-ci sont en effet normalement estimées et une expression approximée est connue. Leur prise en compte directe nécessiterait l'utilisation de formalismes plus généraux tels que les IQC. Ceci permettrait en outre de considérer le caractère variant de certains paramètres, voire les saturations.

Concernant ces dernières, la méthode de synthèse proposée ne permet pas de les prendre en compte directement, par exemple par l'ajout d'un correcteur anti-windup. Il pourrait cependant être envisagé de synthétiser les différents correcteurs de manière simultanée grâce aux outils de synthèse \mathcal{H}_∞ structurée multi-objectifs, tel que proposé dans [Biannic et Apkarian 2011].

Enfin, d'autres stratégies de synthèse robuste de la boucle externe pourraient être envisagées, comme celles issues de la commande adaptative afin de prendre

en compte des cas de pannes ou de fautes. Il serait d'ailleurs intéressant de les comparer avec les résultats obtenus par synthèse \mathcal{H}_∞ .

Sur le plan applicatif

Tout d'abord, la classe de systèmes pour laquelle la méthode de synthèse a été appliquée s'est limitée aux avions rigides évoluant à basse altitude. Il pourrait être intéressant de tester la méthode sur d'autres systèmes, incluant par exemple des modes flexibles qui peuvent être pris en compte lors de la synthèse \mathcal{H}_∞ de la boucle externe.

Pour les applications proposées dans ce mémoire, les mesures (et notamment les variables d'état) ont été considérées connues et accessibles, sans pour autant modéliser précisément d'éventuelles dynamiques et retards dus aux capteurs, ou même envisager la possibilité d'implanter un observateur. Ceci pourrait d'une part entraîner d'autres problématiques non prises en compte dans ces travaux, et d'autre part permettre d'affiner les résultats et rendre les simulations encore plus proches de la réalité.

Étant donné que l'approche de synthèse proposée vise à donner des alternatives aux techniques de séquençement de gains et aux méthodes classiques de linéarisation entrée-sortie, il pourrait être également intéressant de la comparer à ces dernières (les avantages illustrés dans ce mémoire s'avèrent dans un premier temps uniquement pratiques, même si on peut s'attendre également à un gain en performance majeur vis-à-vis de l'inversion dynamique classique notamment).

Enfin, l'implantation du correcteur longitudinal synthétisé au chapitre 7 sur le système réel reste à faire. Elle devrait permettre de valider davantage encore la pertinence de la méthode développée.

PUBLICATIONS

ARTICLE DE REVUE SCIENTIFIQUE

- Lesprier, J., Biannic, J-M. et Roos, C. Nonlinear Structured \mathcal{H}_∞ controllers for parameter-dependent uncertain systems with application to aircraft landing. Soumis à *Aerospace Science and Technology*, 2015.

ARTICLES DE CONFÉRENCES INTERNATIONALES

- Lesprier, J., Biannic, J-M. et Roos, C. Nonlinear Structured \mathcal{H}_∞ controllers for parameter-dependent uncertain systems with application to aircraft landing. Dans *Proceedings of the IEEE Multiconference on Systems and Control*, pages 433-438, Antibes, France, 2014.
- Lesprier, J., Roos, C. et Biannic, J-M. Improved μ upper bound computation using the μ -sensitivities. Dans *Proceedings of the 8th IFAC Symposium on Robust Control Design*, pages 214-219, Bratislava, Slovak Republic, 2015.
- Lesprier, J., Biannic, J-M. et Roos, C. Modeling and robust nonlinear control of a fixed-wing UAV. A paraître dans *Proceedings of the IEEE Multiconference on Systems and Control*, Sydney, Australia, 2015.
- Lesprier, J., Roos, C. et Biannic, J-M. Efficient computation of the multiplier based μ upper bound on large frequency intervals. A paraître dans *Proceedings of the 54th IEEE Conference on Decision and Control*, Osaka, Japan, 2015.

BIBLIOGRAPHIE

- Ackermann, J. Multi-model approaches to robust control system design. Dans *Uncertainty and Control*, volume 70 de *Lecture Notes in Control and Information Sciences*, pages 108–130. Springer Berlin Heidelberg, 1985. (Cité page 104.)
- Adams, R.J. et Banda, S.S. Robust flight control design using dynamic inversion and structured singular value synthesis. *IEEE Transactions on Control Systems Technology*, 1(2), p. 80–92, 1993. (Cité pages 2 et 71.)
- Alazard, D., Cumer, C., Apkarian, P., Gauthier, M., et Ferreres, G. *Robustesse et commande optimale*. Cepadues Editions, Toulouse, France, 1999. (Cité page 83.)
- Anderson, B.D.O. A system theory criterion for positive real matrices. *SIAM Journal of Control*, 5(2), p. 171–182, 1967. (Cité page 34.)
- Apkarian, P. et Gahinet, P. A convex characterization of gain-scheduled \mathcal{H}_∞ controllers. *IEEE Transactions on Automatic Control*, 40(5), p. 2021–2025, 1995. (Cité page 2.)
- Apkarian, P. et Noll, D. Nonsmooth H_∞ synthesis. *IEEE Transactions on Automatic Control*, 51(1), p. 71–86, 2006. (Cité pages 83, 87 et 100.)
- Apkarian, P. et Noll, D. Nonsmooth optimization for multiband frequency domain control design. *Automatica*, 43(3), p. 724–731, 2007. (Cité page 87.)
- Balas, G., Doyle, J., Glover, K., Packard, A., et Smith, R. μ -analysis and synthesis toolbox : User’s Guide, 1991. (Cité page 104.)
- Bates, D. et Mannchen, T. Improved computation of mixed μ bounds for flight control law robustness analysis. *Proceedings of the Institution of Mechanical Engineers, Part I : Journal of Systems and Control Engineering*, 218(8), p. 609–619, 2004. (Cité page 24.)
- Beck, C. et Doyle, J. Mixed μ upper bound computation. Dans *Proceedings of the 31st IEEE Conference on Decision and Control*, pages 3187–3192, Tucson, AZ, USA, December 1992. (Cité pages 3 et 26.)

- Becker, G., Packard, A., Philbrick, D., et Balas, G. Control of parametrically-dependent linear systems : A single quadratic Lyapunov approach. Dans *Proceedings of the American Control Conference*, San Francisco, CA, USA, 1993. (Cité page 2.)
- Biannic, J.-M. et Apkarian, P. A convex characterization of self-scheduled \mathcal{H}_2 controllers. Dans *Proceedings of the IFAC Conference on System Structure and Control*, Nantes, France, 1995. (Cité page 2.)
- Biannic, J.-M. et Apkarian, P. Anti-windup design via nonsmooth multi-objective \mathcal{H}_∞ optimization. Dans *Proceedings of the American Control Conference*, San Francisco, CA, USA, 2011. (Cité pages 99 et 184.)
- Biannic, J.-M., Burlion, L., et de Plinval, H. Robust control design over large flight envelopes : a promising approach for aerial robotics. *AerospaceLab*, 8, p. 1–8, 2014. <http://www.aerospacelab-journal.org/aerial-robotics>. (Cité pages 2 et 71.)
- Biannic, J.-M. et Ferreres, G. Efficient computation of a guaranteed robustness margin. Dans *Proceedings of the 16th IFAC World Congress*, Prague, Czech Republic, 2005. (Cité page 28.)
- Biannic, J.-M. et Roos, C. Flare control law design via multi-channel \mathcal{H}_∞ synthesis : Illustration on a nonlinear aircraft benchmark. Dans *Proceedings of the American Control Conference*, pages 1303–1308, Chicago, IL, USA, 2015. (Cité pages 119 et 127.)
- Bing-Yu, Z. et Blaise, M. Robustness analysis of dynamic inversion control laws applied to nonlinear aircraft pitch-axis models. *Nonlinear Analysis Theory, Methods and applications*, 32(4), p. 501–532, 1998. (Cité page 93.)
- Boiffier, J.-L. *The dynamics of flight : the equations*. Dynamics of Flight Series. Wiley, 1998. ISBN 9780471942375. (Cité pages 116, 119, 131 et 142.)
- Braatz, R. et Morari, M. μ -sensitivities as an aid for robust identification. Dans *Proceedings of the American Control Conference*, pages 231–236, Boston, MA, USA, 1991. (Cité pages 41, 42 et 43.)
- Braatz, R., Young, P., Doyle, J., et Morari, M. Computational complexity of μ calculation. *IEEE Transactions on Automatic Control*, 39(5), p. 1000–1002, 1994. (Cité pages 3 et 19.)
- Brito, A. et Kim, J. Application of a two-step calculation of the real structured singular value to the science mode of the LISA missions. Dans *Proceedings of the*

- AIAA Guidance, Navigation and Control Conference*, Toronto, Canada, 2010. (Cité page 24.)
- Burke, J., Henrion, D., Lewis, A., et Overton, M. HIFOO - a Matlab package for fixed-order controller design and \mathcal{H}_∞ optimization. Dans *Proceedings of the 5th IFAC Symposium on Robust Control Design*, Toulouse, France, 2006. (Cité page 87.)
- Celis, O., Cuyt, A., et Verdonk, B. Rational approximation of vertical segments. *Numerical Algorithms*, 45(1-4), p. 375–388, 2007. (Cité page 16.)
- Chen, G. et Sugie, T. New upper bound of the real μ based on the parameter dependent multiplier. Dans *Proceedings of the 35th IEEE Conference on Decision and Control*, pages 1293–1294, Kobe, Japan, 1996. (Cité page 30.)
- Chen, G. et Sugie, T. An upper bound of μ based on the parameter dependent multipliers. Dans *Proceedings of the American Control Conference*, pages 2604–2608, Albuquerque, NM, USA, 1997. (Cité page 30.)
- Chen, G. et Sugie, T. μ -analysis and synthesis of state feedback systems based on multipliers and LMI's. Dans *Proceedings of the American Control Conference*, pages 537–541, Philadelphia, PA, USA, 1998. (Cité page 30.)
- Chen, K-C., Wang, C-S., et Yen, C-C. Numerical algorithms for the largest structured singular value of a μ -synthesis control system. *Taiwanese Journal of Mathematics*, 14(3A), p. 973–998, 2010. (Cité page 27.)
- Daafouz, J. On inexact LPV control design of continuous-time polytopic systems. *IEEE Transactions on Automatic Control*, 53(7), p. 1674–1678, 2008. (Cité page 2.)
- Dailey, R. A new algorithm for the real structured singular value. Dans *Proceedings of the American Control Conference*, pages 3036–3040, San Diego, CA, USA, 1990. (Cité page 20.)
- De Gaston, R. et Safonov, M. Exact calculation of the multiloop stability margin. *IEEE Transactions on Automatic Control*, 33(2), p. 156–171, 1988. (Cité page 20.)
- Demourant, F. New algorithmic approach based on integral quadratic constraints. Dans *Proceedings of the European Control Conference*, pages 359–364, Zurich, Switzerland, 2013. (Cité page 107.)
- Douglas, J. et Athans, M. The calculation of μ -sensitivities. Dans *Proceedings of the American Control Conference*, pages 437–447, Seattle, WA, USA, 1995. (Cité pages 44 et 45.)

- Doyle, J. Analysis of feedback systems with structured uncertainties. *IEEE Proceedings of Control Theory and Applications, Part D*, 129(6), p. 242–250, 1982. (Cité pages 2, 13 et 18.)
- Doyle, J. Structured uncertainty in control system design. Dans *Proceedings of the 24th IEEE Conference on Decision and Control*, pages 260–265, Fort Lauderdale, FL, USA, 1985. (Cité page 104.)
- Doyle, J. C., Glover, K., Khargonekar, P. P., et Francis, B. A. State-space solutions to standard \mathcal{H}_2 and \mathcal{H}_∞ control problems. *IEEE Transactions on Automatic Control*, 34, p. 831–847, 1989. (Cité pages 83 et 85.)
- Döll, C., Bérard, C., Knauf, A., et Biannic, J-M. LFT modeling of the 2-DOF longitudinal nonlinear aircraft behaviour. Dans *Proceedings of the 9th IEEE Symposium on Computer-Aided Control System Design*, pages 864–869, San Antonio, TX, USA, 2008. (Cité page 16.)
- Fabrizi, A., Roos, C., et Biannic, J-M. A detailed comparative analysis of μ lower bound algorithms. Dans *Proceedings of the European Control Conference*, pages 220–226, Strasbourg, France, 2014. (Cité pages 20, 35, 36 et 41.)
- Fan, M., Tits, A., et Doyle, J. Robustness in the presence of mixed parametric uncertainty and unmodeled dynamics. *IEEE Transactions on Automatic Control*, 36(1), p. 25–38, 1991. (Cité pages 3 et 26.)
- Ferreres, G. et Biannic, J-M. Reliable computation of the robustness margin for a flexible aircraft. *Control Engineering Practice*, 9(12), p. 1267–1278, 2001. (Cité pages 3, 36 et 41.)
- Ferreres, G. et Fromion, V. Computation of the robustness margin with the skewed μ tool. *Systems & Control letters*, 32, p. 193–202, 1997. (Cité page 28.)
- Fliess, M. A new approach to the noninteracting control problem in nonlinear system theory. Dans *Proceedings of the 23rd IEEE Conference on Decision and Control*, pages 192–200, Monticello, VA, USA, 1985. (Cité page 71.)
- Franco, A.L.D., Bourlès, H., De Pieri, E.R., et Guillard, H. Robust nonlinear control associating robust feedback linearization and \mathcal{H}_∞ control. *IEEE Transactions on Automatic Control*, 51(7), p. 1200–1207, 2006. (Cité page 93.)
- Fransson, C-M. A bisection algorithm for the mixed μ upper bound and its supremum. Dans *Proceedings of the American Control Conference*, pages 2665–2670, Boston, MA, USA, 2004. (Cité page 27.)

- Freudenberg, J.S., Looze, D.P., et Cruz, J.B. Robustness analysis using singular value sensitivities. *International Journal of Control*, 35(1), p. 95–116, 1982. (Cité page 45.)
- Fu, M. et Barabanov, N. Improved upper bounds of the structured singular value. Dans *Proceedings of the 34th IEEE Conference on Decision and Control*, pages 3115–3120, New Orleans, LA, USA, 1995. (Cité page 30.)
- Fu, M. et Barabanov, N. Improved upper bounds for the mixed structured singular value. *IEEE Transactions on Automatic Control*, 42(10), p. 1447–1452, 1997. (Cité pages 3, 5, 30, 38, 41, 53, 56, 57 et 64.)
- Gahinet, P. et Apkarian, P. A linear matrix inequality approach to \mathcal{H}_∞ control. *International Journal of Robust and Nonlinear Control*, 4, p. 421–448, 1994. (Cité pages 56 et 85.)
- Gahinet, P. et Apkarian, P. Decentralized and fixed-structure \mathcal{H}_∞ control in MATLAB. Dans *50th IEEE Conference on Decision and Control*, pages 8205–8210, Orlando, FL, USA, 2011a. (Cité page 87.)
- Gahinet, P. et Apkarian, P. Structured \mathcal{H}_∞ synthesis in MATLAB. Dans *Proceedings of the 18th IFAC World Congress*, pages 1435–1440, Milan, Italy, 2011b. (Cité page 87.)
- Gahinet, P., Apkarian, P., Chilali, M., et Feron, E. Affine parameter-dependent Lyapunov functions and real parameter uncertainty. Dans *Proceedings of the 3rd European Control Conference*, pages 2262–2267, Rome, Italy, 1995. (Cité page 30.)
- Gumussoy, S. et Overton, M. Fixed-order \mathcal{H}_∞ controller design via HIFOO, a specialized nonsmooth optimization package. Dans *Proceedings of the American Control Conference*, pages 2750–2754, Seattle, WA, USA, 2008. (Cité page 100.)
- Halton, M., Hayes, M., et Iordanov, P. State-space μ analysis for an experimental drive-by-wire vehicle. *International Journal of Robust and Nonlinear Control*, 18, p. 975–992, 2008. (Cité pages 24 et 28.)
- Hardier, G., Roos, C., et Seren, C. Creating sparse rational approximations for linear fractional representations using genetic programming. Dans *Proceedings of the 3rd IFAC International Conference on Intelligent Control and Automation Science*, pages 232–237, Chengdu, China, 2013a. (Cité page 16.)
- Hardier, G., Roos, C., et Seren, C. Creating sparse rational approximations for linear fractional representations using surrogate modeling. Dans *Proceedings of the 3rd IFAC International Conference on Intelligent Control and Automation Science*, pages 238–243, Chengdu, China, 2013b. (Cité page 16.)

- Hayes, M., Bates, D., et Postlethwaite, I. New tools for computing tight bounds on the real structured singular value. *AIAA Journal of Guidance, Control and Dynamics*, 24(6), p. 1204–1213, 2001. (Cité page 24.)
- Hernandez Lopezomoza, M.A. *A generalized framework for robust nonlinear compensation : Application to an atmospheric reentry control problem*. PhD thesis, Institut Supérieur de l'Aéronautique et de l'Espace, Toulouse, France, 2012. (Cité page 71.)
- Holland, R., Young, P., et Zhu. Development of a skew μ upper bound. *International Journal of Robust and Nonlinear Control*, 15, p. 905–921, 2005. (Cité page 28.)
- Iordanov, P. et Halton, M. Computation of the real structured singular value via pole migration. *International Journal of Robust and Nonlinear Control*, 2014. (Cité page 23.)
- Iordanov, P., Hayes, M., et Halton, M. On μ -analysis and synthesis for systems subject to real uncertainty. Dans *Proceedings of the 7th European Control Conference*, pages 420–425, Cambridge, United Kingdoms, 2003. (Cité page 23.)
- Isidori, A. *Nonlinear Control Systems*, volume 1 de *Communications and Control Engineering*. Springer, 1985. (Cité pages 2 et 71.)
- Isidori, A., Krener, A., Gori-Giorgi, C., et Monaco, S. Nonlinear decoupling via feedback : A differential geometric approach. *IEEE Transactions on Automatic Control*, 26(2), p. 331 – 345, 1981. ISSN 0018-9286. (Cité page 71.)
- Ito, D., Georgie, J., Valasek, J., et Ward, D. *Reentry vehicle flight controls design guidelines : Dynamic inversion*. PhD thesis, Texas A&M University, Texas, USA, 2002. (Cité page 71.)
- Iwasaki, T. et Hara, S. Well-posedness of feedback systems : insights into exact robustness analysis and approximate computations. *IEEE Transactions on Automatic Control*, 43(5), p. 619–630, 1998. (Cité pages 31 et 33.)
- Jönsson, U., Kao, C.-Y., Megretski, A., et Rantzer, A. *A guide to IQC β : a MATLAB toolbox for robust stability and performance analysis*, 2004. (Cité page 107.)
- Jönsson, U. et Rantzer, A. Systems with uncertain parameters - time variations with bounded derivatives. *International Journal of Robust and Nonlinear Control*, 6, p. 969–982, 1996. (Cité page 107.)
- Kara Mohamed, M. et Lanzon, A. Design and control of novel tri-rotor UAV. Dans *Proceedings of the 2012 UKACC International Conference on Control*, pages 304–309, Cardiff, Wales, 2012. (Cité pages 2 et 71.)

- Kara Mohamed, M. et Lanzon, A. Effect of unmodelled actuator dynamics on feedback linearised systems and a two stage feedback linearisation method. Dans *Proceedings of the 52nd IEEE Conference on Decision and Control*, pages 841–846, Florence, Italy, 2013. (Cité pages 71 et 93.)
- Kasimbeyli, R., Ostun, O., et Rubinov, A. The modified subgradient algorithm based on feasible values. *Optimization*, 58(5), p. 535–560, 2009. (Cité page 24.)
- Kato, T. *Perturbation Theory for Linear Operators*. Springer-Verlag, 1976. (Cité page 45.)
- Katz, J. et Plotkin, A. *Low-Speed Aerodynamics : From wing theory to panel methods*. 1991. (Cité page 163.)
- Kim, J., Bates, D.G., et Postlethwaite, I. A geometrical formulation of the μ lower bound problem. *IET, Control Theory & Applications*, 3(4), p. 465–472, 2008. (Cité pages 25 et 35.)
- Köroğlu, Hakan et Scherer, Carsten W. Robust performance analysis for structured linear time-varying perturbations with bounded rates-of-variation. *IEEE Transactions on Automatic Control*, 52, p. 197–211, 2007. (Cité page 107.)
- Lemos, R., Simões, A., et Apkarian, P. A non-smooth lower bound on v . *International Journal of Robust and Nonlinear Control*, 13, p. 715–734, 2014. (Cité page 25.)
- Lesprier, J., Biannic, J-M., et Roos, C. Nonlinear structured \mathcal{H}_∞ controllers for parameter-dependent uncertain systems with application to aircraft landing. Dans *Proceedings of the IEEE Multiconference on Systems and Control*, pages 433–438, Antibes, France, 2014. (Cité page 5.)
- Lesprier, J., Biannic, J-M., et Roos, C. Modeling and robust nonlinear control of a fixed-wing UAV. Dans *Proceedings of the IEEE Multiconference on Systems and Control*, Sydney, Australia, 2015a. (Cité page 6.)
- Lesprier, J., Roos, C., et Biannic, J-M. Efficient computation of the multiplier based μ upper bound on large frequency intervals. Dans *Proceedings of the 54th IEEE Conference on Decision and Control*, Osaka, Japan, 2015b. (Cité page 5.)
- Lesprier, J., Roos, C., et Biannic, J-M. Improved μ upper bound computation using the μ -sensitivities. Dans *Proceedings of the IFAC ROCOND*, pages 214–219, Bratislava, Slovak Republic, 2015c. (Cité page 5.)
- Looye, G. et Joos, H.-D. Design of autoland controller functions with multiobjective optimization. *Journal of Guidance, Control and Dynamics*, 29(2), p. 475–484, 2006. (Cité page 147.)

- Ly, J., Chiang, R., Goh, K-C., et Safonov, M. LMI multiplier K_m/μ -analysis of the Cassini spacecraft. *International Journal of Robust and Nonlinear Control*, 8, p. 155–168, 1998. (Cité page 31.)
- Ly, J., Safonov, M., et Chiang, R. Real/Complex multivariable stability margin computation via generalized Popov multiplier - LMI approach. Dans *Proceedings of the American Control Conference*, pages 425–429, Baltimore, MD, USA, 1994. (Cité page 31.)
- Magni, J.-F. Linear Fractional Representation Toolbox (version 2.0) for use with Matlab. Available at <http://w3.onera.fr/smac/>, 2006. (Cité page 17.)
- Magni, J.-F. et Döll, C. A new lower bound of the mixed structured singular value. Dans *Proceedings of the 2nd Asian Control Conference*, pages 847–850, Seoul, South Korea, 1997. (Cité page 22.)
- Magni, J.-F., Döll, C., Chiappa, C., Frappard, B., et Girouart, B. Mixed μ -analysis for flexible systems. Part 1 : Theory. Dans *Proceedings of the 14th IFAC World Congress*, pages 325–360, Beijing, China, 1999. (Cité page 22.)
- Marcos, A. et Balas, G. Development of Linear-Parameter-Varying models for aircraft. *Journal of Guidance, Control and Dynamics*, 27(2), p. 218–228, 2004. (Cité page 93.)
- Matsuda, T., Kawanishi, M., et Narikiyo, T. Computation of real structured singular value by Stability Feeler. Dans *Proceedings of the Asian Control Conference*, pages 672–677, Hong-Kong, China, 2009. (Cité pages 21 et 34.)
- Matsuda, T. et Mori, T. Stability Feeler : a tool for parametric robust stability analysis and its applications. *IET Control Theory & Applications*, 3(12), p. 1625–1633, 2009. (Cité pages 21 et 34.)
- Megretski, A. et Rantzer, A. System analysis via integral quadratic constraints. *IEEE Transactions on Automatic Control*, 42(6), p. 819–830, 1997. (Cité pages 33 et 107.)
- Meinsma, G., Shrivastava, Y., et Fu, M. Some properties of an upper bound for μ . *IEEE Transactions on Automatic Control*, 41(9), p. 1326–1330, 1996. (Cité page 30.)
- Newlin, M. et Glavaski, S. Advances in the computation of the μ lower bound. Dans *Proceedings of the American Control Conference*, pages 442–446, Seattle, WA, USA, 1995. (Cité page 21.)

- Newlin, M.P. et Young, P.M. Mixed μ problems and branch and bound techniques. *International Journal of Robust and Nonlinear Control*, 7(2), p. 145–164, 1997. (Cité pages 3, 37, 41, 48, 51 et 53.)
- Packard, A. Gain scheduling via linear fractional transformations. *Systems and Control Letters*, 22(2), p. 79–92, 1994. (Cité page 2.)
- Packard, A., Balas, G., Liu, R., et Shin, J.Y. Results on worst-case performance assessment. Dans *Proceedings of the American Control Conference*, pages 2425–2427, Chicago, IL, USA, 2000. (Cité page 22.)
- Packard, A., Fan, M., et Doyle, J. A power method for the structured singular value. Dans *Proceedings of the 27th IEEE Conference on Decision and Control*, pages 2132–2137, Austin, TX, USA, 1988. (Cité page 21.)
- Packard, A. et Pandey, P. Continuity properties of the real/complex structured singular value. *IEEE Transactions on Automatic Control*, 38(3), p. 415–428, 1993. (Cité page 21.)
- Papageorgiou, C. et Glover, K. Robustness analysis of nonlinear dynamic inversion control laws with application to flight control. Dans *Proceedings of the 43rd IEEE Conference on Decision and Control*, pages 3485–3490, Nassau, The Bahamas, 2004. (Cité page 93.)
- Papageorgiou, C. et Glover, K. Robustness analysis of nonlinear flight controllers. *Journal of Guidance, Control and Dynamics*, 28(4), p. 639–648, 2005. (Cité page 93.)
- Poussot-Vassal, C. et Roos, C. Generation of a reduced-order LPV/LFT model from a set of large-scale MIMO LTI flexible aircraft models. *Control Engineering Practice*, 20(9), p. 919–930, 2012. (Cité page 16.)
- Reiner, J., Balas, G., et Garrard, W. Flight control design using robust dynamic inversion and time-scale separation. *Automatica*, 32(11), p. 1493–1504, 1996. (Cité pages 2 et 71.)
- Reiner, J., Balas, G.J., et Garrard, W.L. Robust dynamic inversion for control of highly maneuverable aircraft. *Journal of Guidance, Control and Dynamics*, 18(1), p. 18–24, 1995. (Cité pages 2 et 71.)
- Roos, C. Generation of flexible aircraft LFT models for robustness analysis. Dans *Proceedings of the 6th IFAC Symposium on Robust Control Design*, Haifa, Israel, 2009. (Cité page 16.)

- Roos, C. et Biannic, J-M. Efficient computation of a guaranteed stability domain for a high-order parameter dependent plant. Dans *Proceedings of the American Control Conference*, pages 3895–3900, Baltimore, MD, USA, 2010. (Cité pages 28 et 50.)
- Roos, C. et Biannic, J-M. A detailed comparative analysis of all practical algorithms to compute lower bounds on the structured singular value. *To appear in Control Engineering Practice*, 2015. (Cité page 3.)
- Roos, C., Hardier, G., et Biannic, J-M. Polynomial and rational approximation with the APRICOT Library of the SMAC Toolbox. Dans *Proceedings of the IEEE Multiconference on Systems and Control*, pages 1473–1478, Antibes, France, 2014. (Cité pages 16, 101 et 137.)
- Schumacher, C. et Khargonekar, P. Stability analysis of a missile control system with a dynamic inversion controller. Dans *Proceedings of the American Control Conference*, pages 1732–1736, Philadelphia, PA, USA, 1998. (Cité page 71.)
- Scorletti, G. *Approche unifiée de l'analyse et de la commande des systèmes par optimisation LMI*. PhD thesis, Université Paris Sud - Paris XI, 1997. (Cité page 107.)
- Scorletti, G., Bombois, X., Barenthin, M., et Fromion, V. Improved efficient analysis for systems with uncertain parameters. Dans *Proceedings of the 46th IEEE Conference on Decision and Control*, pages 5038–5043, New Orleans, LA, USA, 2007. (Cité pages 3 et 27.)
- Seiler, P., Packard, A., et Balas, G. A gain-based lower bound algorithm for real and mixed μ problems. *Automatica*, 46, p. 493–500, 2010. (Cité pages 22 et 36.)
- Shahruz, S. M. et Behtash, S. Design of controllers for linear parameter-varying systems by the gain scheduling technique. Dans *Proceedings of the 29th Conference on Decision and Control*, pages 2490–2491, 1990. (Cité page 1.)
- Shamma, J.F. et Athans, M. Guaranteed properties of gain-scheduled control for nonlinear plants. *IEEE Transactions on Automatic Control*, 35(8), p. 898–907, 1990. (Cité page 1.)
- Sideris, A. Elimination of frequency search from robustness tests. *IEEE Transactions on Automatic Control*, 37(10), p. 1635–1640, 1992. (Cité page 28.)
- Sideris, A. et De Gaston, R. Multivariable stability margin calculation with uncertain correlated parameters. Dans *Proceedings of the 25th IEEE Conference on Decision and Control*, pages 766–771, Athens, Greece, 1986. (Cité page 20.)

- Sideris, A. et Sanchez Pena, R. Fast computation of the multivariable stability margin for real interrelated uncertain parameters. *IEEE Transactions on Automatic Control*, 34(12), p. 1272–1276, December 1989. (Cité page 20.)
- Sideris, A. et Sanchez Pena, R. Robustness margin calculation with dynamic and real parametric uncertainty. *IEEE Transactions on Automatic Control*, 35(8), p. 970–974, 1990. (Cité page 20.)
- Sivells, J. et Neely, R. Method for calculating wing characteristics by lifting line theory using nonlinear section lift data. Rapport technique, NACA Technical Note, 1947. (Cité page 162.)
- Slotine, J. et Li, W. *Applied Nonlinear Control*. New Jersey : Prentice Hall, 1991. (Cité pages 2 et 71.)
- SMAC. Systems Modeling Analysis & Control Toolbox. ONERA Research Project, <http://w3.onera.fr/smac>, 2012-2015. (Cité pages ix, 16, 36, 101, 103, 105, 127, 128, 137, 154, 164 et 167.)
- Snell, S. A., Dale, F., et William Garrard, L. Nonlinear inversion flight control for a supermaneuverable aircraft. *Journal of Guidance, Control, and Dynamics*, 15(4), p. 976–984, 1992. (Cité pages 2 et 71.)
- Sparks, A. et Bernstein, D. Reliable state space upper bounds for the peak structured singular value. Dans *Proceedings of the American Control Conference*, pages 2419–2423, Seattle, WA, USA, 1995. (Cité page 31.)
- The MathWorks, Inc. Robust Control Toolbox. <http://www.mathworks.fr/help/toolbox/robust/ref/hinfstruct.html>, 2010. (Cité pages 36 et 104.)
- The MathWorks, Inc. Curve Fitting Toolbox user's guide. <http://www.mathworks.com/products/curvefitting>, 2015a. (Cité page 16.)
- The MathWorks, Inc. Matlab optimization Toolbox user's guide. <https://www.mathworks.com/products/optimization/>, 2015b. (Cité pages 23 et 24.)
- Tierno, J. et Young, P. An improved μ lower bound via adaptive power iteration. Dans *Proceedings of the 31st IEEE Conference on Decision and Control*, pages 3181–3186, Tucson, AZ, USA, 1992. (Cité page 21.)
- Vidyasagar, M. *Nonlinear Systems Analysis*. New Jersey : Prentice Hall, 1993. (Cité pages 2 et 71.)

- Vuillemin, P., Demourant, F., Biannic, J.-M., et Poussot-Vassal, C. Global stability validation of an uncertain large-scale aircraft model. Dans *Proceedings of the IEEE Conference on Control Applications*, pages 152–157, Antibes, France, 2014. (Cité page 107.)
- Yazıcı, A., Karamancioğlu, A., et Kasimbeyli, R. A nonlinear programming technique to compute a tight lower bound for the real structured singular value. *Optimization and Engineering*, 12(3), p. 445–458, 2011. (Cité page 24.)
- Young, P. Controller design with real parametric uncertainties. *International Journal of Control*, 65(3), p. 469–509, 1996. (Cité page 104.)
- Young, P. et Doyle, J. Computation of μ with real and complex uncertainties. Dans *Proceedings of the 29th IEEE Conference on Decision and Control*, pages 1230–1235, Honolulu, Hawaii, 1990. (Cité page 21.)
- Young, P. et Doyle, J. A lower bound for the mixed μ problem. *IEEE Transactions on Automatic Control*, 42(1), p. 123–128, 1997. (Cité pages 21 et 36.)
- Young, P., Newlin, M., et Doyle, J. Practical computation of the mixed μ problem. Dans *Proceedings of the American Control Conference*, pages 2190–2194, Chicago, IL, USA, 1992. (Cité pages 3 et 26.)
- Young, P., Newlin, M., et Doyle, J. Computing bounds for the mixed- μ problems. *International Journal of Robust and Nonlinear Control*, 5(6), p. 573–590, 1995. (Cité pages 3, 26, 41 et 65.)
- Zadeh, L. et Desoer, C. *Linear System Theory*. New York : McGraw Hill, 1963. (Cité page 20.)
- Zames, G. Feedback and optimal sensitivity : model reference transformations, multiplicative seminorms, and approximations. *IEEE Transactions on Automatic Control*, 26(2), p. 301–320, 1981. (Cité page 83.)
- Zhao, Y.-B., Kim, J., et Bates, D. LFT-free μ -analysis of LTI/LPTV systems. Dans *Proceedings of the IEEE International Symposium on Computer-Aided Control System Design*, pages 638–643, Denver, CO, USA, 2011. (Cité page 35.)
- Zhou, K. et Doyle, J. C. *Essentials of robust control*. Prentice-Hall, Inc., 1998. (Cité page 83.)
- Zhou, K., Doyle, J. C., et Glover, K. *Robust and optimal control*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1996. ISBN 0-13-456567-3. (Cité pages 13, 15 et 83.)

Titre Synthèse et validation de lois de commande de vol robustes en présence d'incertitudes paramétriques et de non-linéarités.

Résumé Les méthodes de synthèse et de validation de lois de pilotage utilisées dans le milieu industriel aéronautique sont bien souvent longues et coûteuses à mettre en œuvre. Pourtant, des alternatives pour traiter les larges domaine d'évolution existent, comme l'inversion dynamique robuste. Ce travail de thèse cherche à en corriger les défauts, notamment grâce à une meilleure prise en compte des incertitudes du système et à une rejection des non-linéarités non compensées. Ceci est rendu possible grâce aux techniques de synthèse \mathcal{H}_∞ structurée multi-modèles, qui peuvent considérer plusieurs configurations pire-cas déterminées par des outils d'analyse de robustesse. Une autre contribution est d'ailleurs l'amélioration des techniques permettant d'évaluer la marge de robustesse d'un système LTI, souvent trop conservatives ou nécessitant un temps de calcul prohibitif. Les méthodes proposées sont appliquées au pilotage automatique d'un avion civil et d'un drone à voilure fixe, dont la modélisation est également traitée avec précision.

Mots-clés inversion dynamique robuste, synthèse \mathcal{H}_∞ structurée multi-modèles, variations paramétriques, μ -analyse, incertitudes LTI, applications aéronautiques.

Title Design and validation of robust flight control laws in the presence of parametric uncertainties and nonlinearities.

Abstract Control laws design and validation methods used in the aeronautical industry are generally time-consuming and costly to set up. However, alternatives exist to cope with large operating domains, such as robust dynamic inversion. This thesis work aims at fixing its drawbacks by better considering the system uncertainties, and by rejecting the uncompensated nonlinearities. Thanks to recent advances in structured multimodel \mathcal{H}_∞ design techniques, it is now possible to consider multiple worst-case configurations, determined using robustness analysis tools. Besides, another part of this work is devoted to the improvement of existing techniques to evaluate the robustness margin of an LTI system, which are often too conservative or computationally costly. All these contributions are applied to the control of a civil aircraft and a small fixed-wing UAV, whose modeling is also thoroughly described.

Keywords robust dynamic inversion, structured multimodel \mathcal{H}_∞ synthesis, parametric variations, μ -analysis, LTI uncertainties, aeronautical applications.